ALIBABA CLOUD

阿里云

云原生数据仓库 AnalyticDB PostgreSQL 版

数据接入

文档版本：20210927

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.数据迁移及同步方案综述

AnalyticDB for PostgreSQL 提供了多种数据迁移方案，可满足不同的数据同步或迁移的业务需求，使您可以在不影响业务的情况下，平滑地与各种数据库类型实例之间进行迁移或数据同步，包括 RDS MySQL，PolarDB for MySQL， RDS PostgreSQL、RDS PPAS、MaxCompute、Greenplum Database、以及自建MySQL 、 PostgreSQL 或 Amazon Redshift。除下面所列阿里云方案外，也支持第三方数据同步产品，例如DSG。

AnalyticDB for PostgreSQL支持的各种数据迁移应用场景及相关操作如下：

操作	类型	场景
使用OSS外表高速导入OSS数据	数据迁移	通过OSS外部表将数据在AnalyticDB for PostgreSQL和OSS之间进行导入或者导出。
Dataworks数据集成	数据同步/数据迁移	通过数据集成（Data Integration）服务，实现分钟级延迟的数据同步，支持AnalyticDB for PostgreSQL作为同步的目标端或者源端，同其它多种异构数据源，进行数据的导入或者导出。
使用\COPY命令导入本地数据	数据迁移	通过 `\COPY` 命令，将本地的文本文件数据导入到AnalyticDB for PostgreSQL中。
从RDS PostgreSQL同步至云原生数据仓库AnalyticDB PostgreSQL版	数据同步/数据迁移	通过数据传输服务（DTS）同步RDS PostgreSQL 数据到 AnalyticDB for PostgreSQL。
RDS MySQL数据同步至AnalyticDB PostgreSQL版	数据同步/数据迁移	通过数据传输服务（DTS）同步RDS MySQL数据到AnalyticDB for PostgreSQL。
PolarDB MySQL数据同步至AnalyticDB PostgreSQL	数据同步/数据迁移	通过数据传输服务（DTS）同步PolarDB for MySQL数据到AnalyticDB for PostgreSQL。
从ECS上的自建MySQL同步至AnalyticDB PostgreSQL版	数据同步/数据迁移	通过数据传输服务（DTS）同步ECS自建MySQL数据到AnalyticDB for PostgreSQL
从通过专线、VPN网关或智能接入网关接入的自建MySQL同步至AnalyticDB PostgreSQL版	数据同步/数据迁移	通过数据传输服务（DTS）同步专线/VPN网关/智能网关接入的云下自建MySQL数据同步至AnalyticDB for PostgreSQL
从RDS SQL Server同步至云原生数据仓库AnalyticDB PostgreSQL	数据同步/数据迁移	通过数据传输服务（DTS）同步RDS SQL Server数据到AnalyticDB for PostgreSQL。
Amazon Redshift应用和数据迁移至AnalyticDB PostgreSQL	第三方迁移	通过Amazon S3和阿里云OSS将Amazon Redshift的数据导入到AnalyticDB for PostgreSQL中。
rds_dbsync迁移/同步MySQL数据到AnalyticDB for PostgreSQL	数据同步/数据迁移	通过开源项目rds_dbsync的mysql2pgsql工具将本地MySQL中的表同步到AnalyticDB for PostgreSQL中。
rds_dbsync迁移/同步PostgreSQL数据到AnalyticDB PostgreSQL	数据同步/数据迁移	通过开源项目rds_dbsync的pgsql2pgsql工具将AnalyticDB for PostgreSQL/Greenplum Database/PostgreSQL/PPAS中的表同步到AnalyticDB for PostgreSQL中。

1. 数据迁移：是指将各种数据库实例或者本地存储的数据迁移到AnalyticDB for PostgreSQL。

2. 数据同步：是指将其他数据库中的数据实时同步到AnalyticDB for PostgreSQL。

2.通过实时计算Flink版读取云原生数据仓库AnalyticDB PostgreSQL

本文介绍如何通过阿里云实时计算Flink版实时读取云原生数据仓库AnalyticDB PostgreSQL（以下简称ADB PG版，原分析型数据库PostgreSQL版）数据，包括版本限制、语法示例、创建和运行Flink作业、WITH参数、CACHE参数、类型映射和参数支持等。

版本限制

- 创建3.6.0及以上版本实时计算集群。
- 创建6.0版本ADB PG集群（实时计算集群和ADB PG版实例需要位于同一VPC下，且ADB PG版实例的白名单规则允许Flink集群网段访问）。

语法示例

```
CREATE TABLE dim_adbpg(
    id int,
    username varchar,
    INDEX(id)
) with(
    type='custom',
    tableFactoryClass='com.alibaba.blink.customersink.ADBPGCustomSourceFactory',
    url='jdbc:postgresql://内网连接串/databasename',
    tableName='tablename',
    userName='username',
    password='password',
    joinMaxRows='100',
    maxRetryTimes='1',
    connectionMaxActive='5',
    retryWaitTime='100',
    targetSchema='public',
    caseSensitive='0',
    cache='LRU',
    cacheSize='1000',
    cacheTTLMs='10000',
    cacheReloadTimeBlackList='2017-10-24 14:00 -> 2017-10-24 15:00',
    partitionedJoin='true'
);

-- join时需要指定在代码中加入维表标识 FOR SYSTEM_TIME AS OF PROCTIME()
INSERT INTO print_sink
SELECT R.c1, R.a2, R.a3, R.a4, R.a5, R.a6, R.a6, R.a8, R.a9, R.a10, R.a11, R.a13, T.username
FROM s_member_cart_view AS R
left join
dim_adbpg FOR SYSTEM_TIME AS OF PROCTIME() AS T
on R.c1 = T.id;
```

WITH参数

参数名	参数含义	备注
url	ADBPG连接地址	必填，需要填写格式为jdbc:postgresql://<ADBPG内网连接串>/databaseName 的内网连接地址。
type	表类型	必填。
tableName	ADBPG源表名	必填，填写维表对应的ADBPG数据仓库中的表名。
userName	ADBPG用户名	必填。
password	ADBPG密码	必填。
joinMaxRows	左表一条记录连接右表的最大记录数	非必填，表示在一对多连接时，左表一条记录连接右表的最大记录数（默认值为1024）。在一对多连接的记录数过多时，可能会极大地影响流任务的性能，因此您需要增大Cache的内存（cacheSize限制的是左表key的个数）。
maxRetryTimes	单次SQL失败后重试次数	非必填，实际执行时，可能会因为各种因素造成执行失败，比如网络或者IO不稳定，超时等原因，ADBPG维表支持SQL执行失败后自动重试，用maxRetryTimes参数可以设定重试次数。默认值为3。
connectionMaxActive	连接池最大连接数	非必填，ADBPG维表中内置连接池，设置合理的连接池最大连接数可以兼顾效率和安全性，默认值为5。
retryWaitTime	重试休眠时间	非必填，每次SQL失败重试之间的sleep间隔，单位ms，默认值100。
targetSchema	查询的ADBPG schema	非必填，默认值public。

参数名	参数含义	备注
caseSensitive	是否大小写敏感	非必填，默认值0，即不敏感；填1可以设置为敏感。

CACHE参数

参数名	参数含义	备注
cache	缓存策略	目前ADB PG版支持以下三种缓存策略： • None（默认值）：无缓存。 • LRU：缓存维表里的部分数据。源表来一条数据，系统会先查找Cache，如果没有找到，则去物理维表中查询。 • ALL：缓存维表里的所有数据。在Job运行前，系统会将维表中所有数据加载到Cache中，之后所有的维表查询都会通过Cache进行。如果在Cache中无法找到数据，则KEY不存在，并在Cache过期后重新加载一遍全量Cache。
cacheSize	设置LRU缓存的最大行数	非必填，默认为10000行。
cacheTTLMs	缓存更新时间间隔。系统会根据您设置的缓存更新时间间隔，重新加载一次维表中的最新数据，保证源表能JOIN到维表的最新数据。	非必填，单位为毫秒。默认不设置此参数，表示不重新加载维表中的新数据。
cacheReloadTimeBlackList	更新时间黑名单。在缓存策略选择为ALL时，启用更新时间黑名单，防止在此时间内做Cache更新（例如双11场景）。	非必填，默认空，格式为 **'2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00'** 。其中分割符使用情况如下： • 用逗号（,）来分隔多个黑名单。 • 用箭头（->）来分割黑名单的起始结束时间。

参数名	参数含义	备注
partitionedJoin	是否开启partitionedJoin。在开启partitionedJoin优化时，主表会在关联维表前，先按照Join KEY进行Shuffle，这样做有以下优点： • 在缓存策略为LRU时，可以提高缓存命中率。 • 在缓存策略为ALL时，节省内存资源，因为每个并发只缓存自己并发所需要的数据。	非必填，默认情况下为false，表示不开启partitionedJoin。

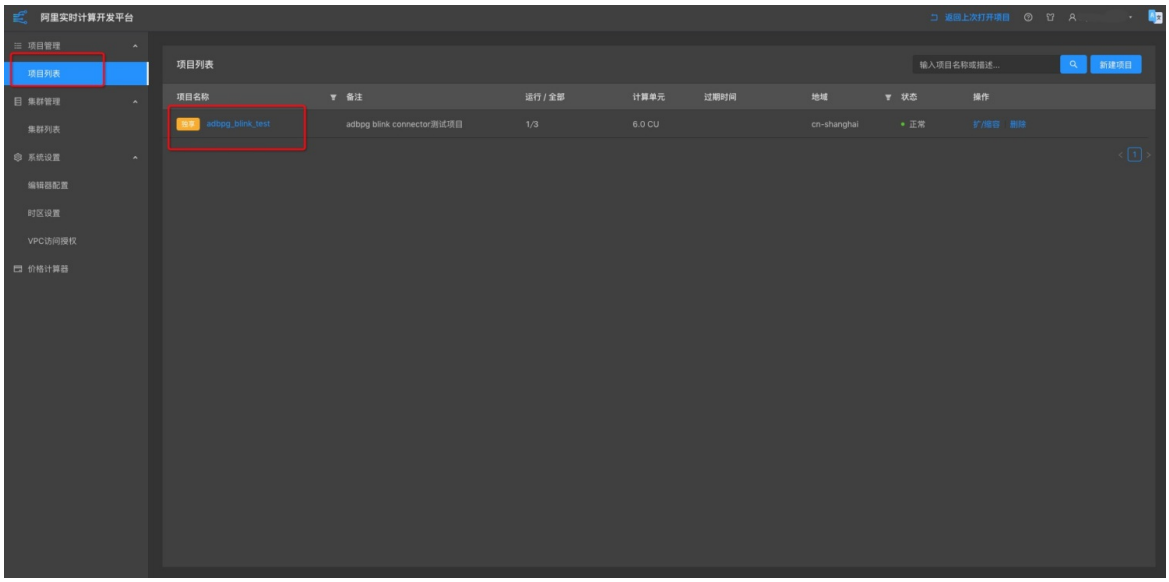
类型映射

实时计算字段类型	ADB PG版字段类型
BOOLEAN	BOOLEAN
TINYINT	SMALLINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
DOUBLE	DOUBLE PRECISION
VARCHAR	TEXT
DATETIME	TIMESTAMP
DATE	DATE
FLOAT	REAL
DECIMAL	DOUBLE PRECISION

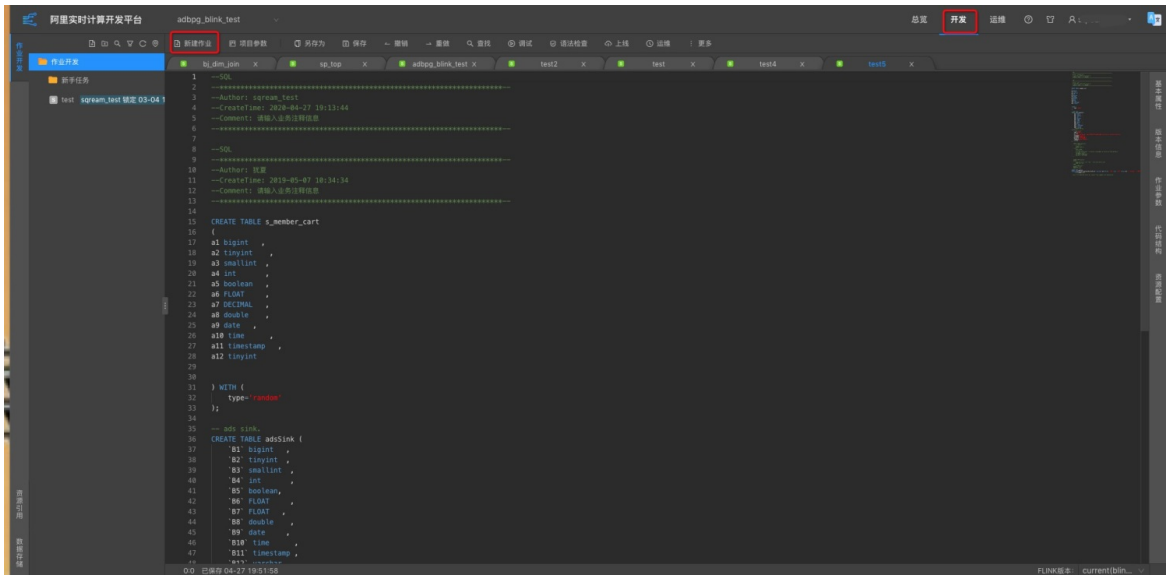
实时计算字段类型	ADB PG版字段类型
TIME	TIME
TIMESTAMP	TIMESTAMP

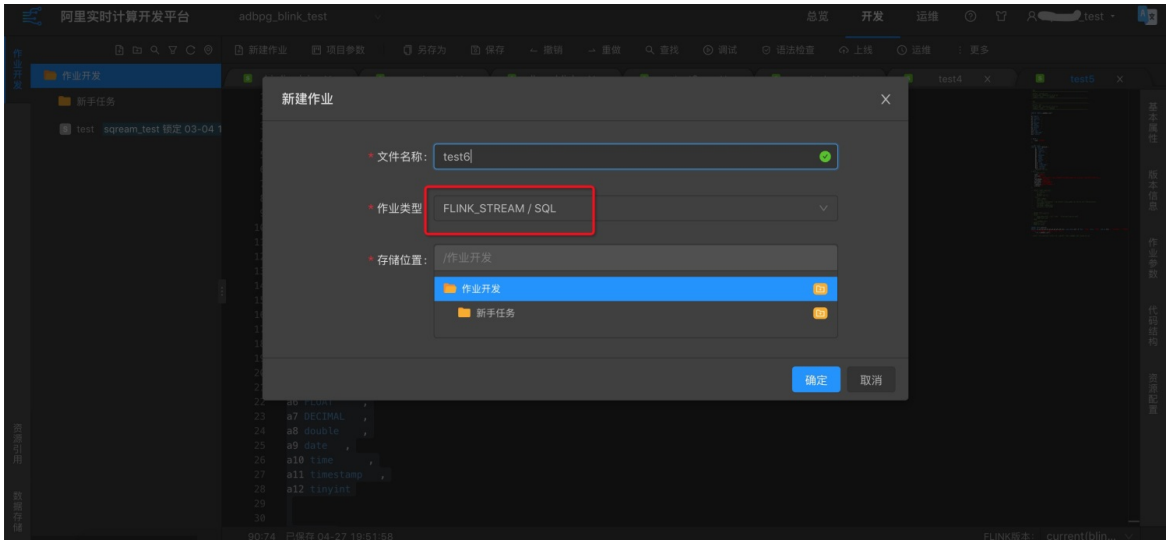
创建和运行Flink作业

1. 登录实时计算控制台，在页面顶部菜单栏上，鼠标悬停在用户头像上，单击项目管理。在项目管理>项目列表页面，单击项目名进入自己创建的项目。

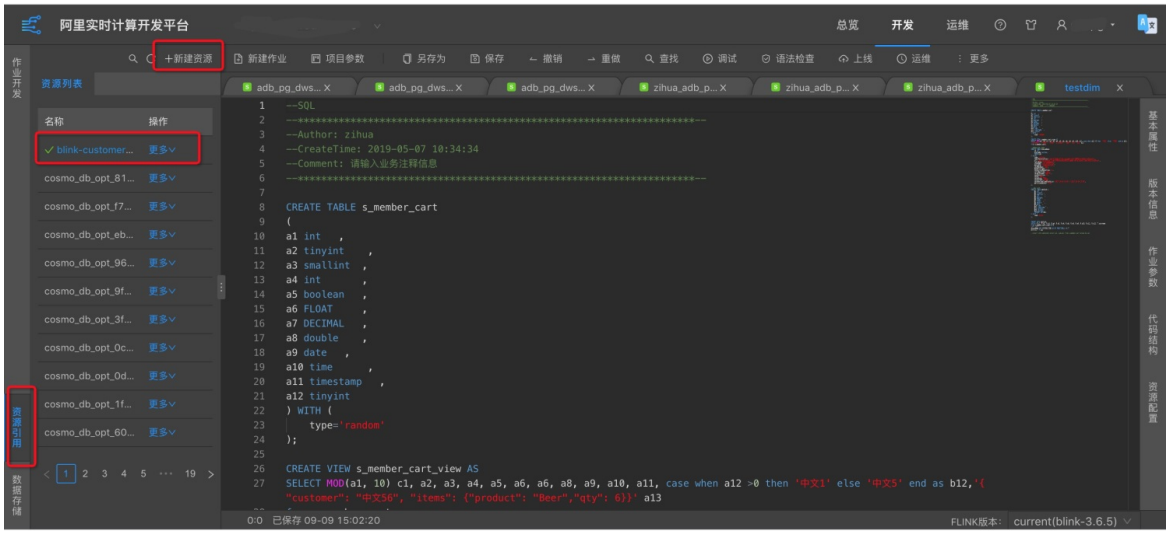


2. 单击开发>新建作业，创建数据写入的Flink SQL作业。



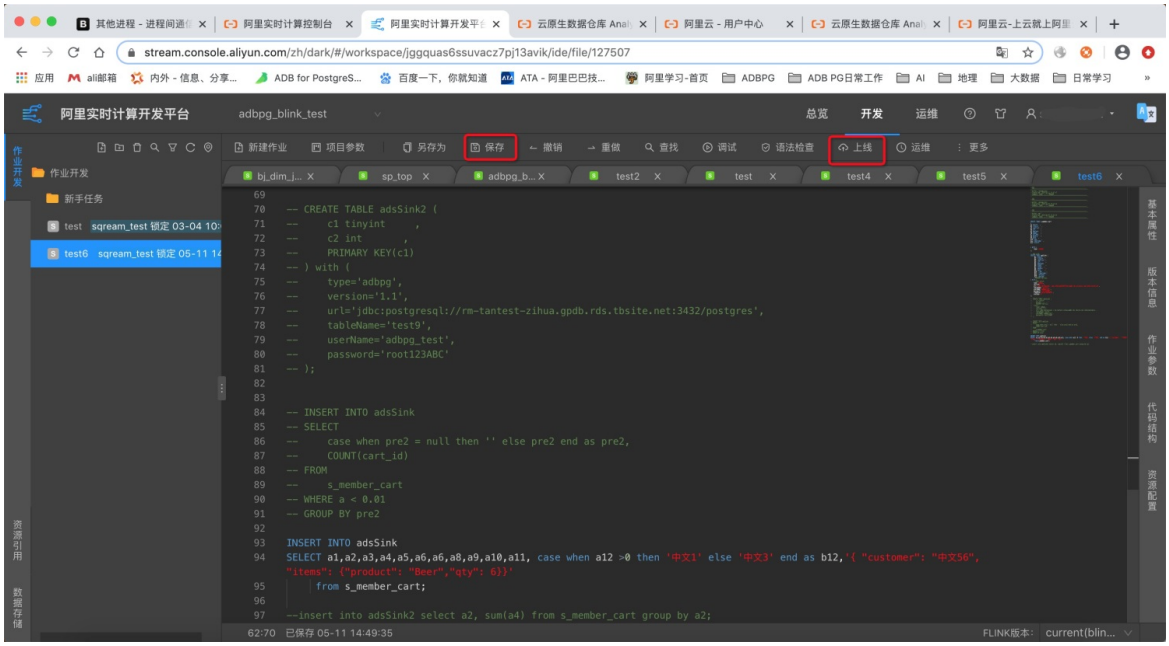


3. 目前采用Flink自定义维表的方式支持读取ADB PG版目标表数据，使用自定义维表功能上线前需要在资源引用界面上传及引用.jar包，编写完作业后点击资源引用>新建资源>上传JAR包>更多>引用。



通过以下链接下载jar包：下载JAR包。

4. 完成作业开发后，依次点击保存、上线，即可上线该任务。



5. 继续点击运维，启动对应项目即可启动任务。

代码示例

这里给出读取ADB PG版数据打印到Flink日志中的Flink SQL示例：

```
--SQL
--********************************************************************--
--Author: zihua
--CreateTime: 2019-09-07 10:34:34
--********************************************************************--

CREATE TABLE s_member_cart
(
    a1 int,
    a2 tinyint  ,
    a3 smallint ,
    a4 int,
    a5 boolean,
    a6 FLOAT    ,
    a7 DECIMAL  ,
    a8 double,
    a9 date  ,
    a10 time    ,
    a11 timestamp ,
    a12 tinyint
) WITH (
    type='random'
);

CREATE VIEW s_member_cart_view AS
SELECT MOD(a1, 10) c1, a2, a3, a4, a5, a6, a6, a8, a9, a10, a11, case when a12 >0 then 'test1' else 'test5' end as
b12,'{ "customer": "中文56", "items": {"product": "Beer","qty": 6}}' a13
FROM s_member_cart;

--adbpg dim index
CREATE TABLE dim_adbpg(
    id int
```

```
    id int,
    username varchar,
    INDEX(id)
) with(
    type='custom',
    tableFactoryClass='com.alibaba.blink.customersink.ADBPGCustomSourceFactory',
    url='jdbc:postgresql://内网连接串/databasename',
    tableName='tablename',
    userName='username',
    password='password',
    joinMaxRows='100',
    maxRetryTimes='1',
    connectionMaxActive='5',
    retryWaitTime='100',
    targetSchema='public',
    caseSensitive='0',
    cache='LRU',
    cacheSize='1000',
    cacheTTLMs='10000',
    cacheReloadTimeBlackList='2017-10-24 14:00 -> 2017-10-24 15:00',
    partitionedJoin='true'
);

-- ads sink.
CREATE TABLE print_sink (
    B1 int,
    B2 tinyint ,
    B3 smallint ,
    B4 int,
    B5 boolean,
    B6 FLOAT   ,
    B7 FLOAT  ,
    B8 double,
    B9 date  ,
    B10 time   ,
    B11 timestamp ,
    B12 varchar,
    B15 varchar,
    PRIMARY KEY(B1)
) with (
    type='print'
);


INSERT INTO print_sink
SELECT R.c1, R.a2, R.a3, R.a4, R.a5, R.a6, R.a6, R.a8, R.a9, R.a10, R.a11, R.a13, T.username
FROM s_member_cart_view AS R
left join
dim_adbpg FOR SYSTEM_TIME AS OF PROCTIME() AS T
on R.c1 = T.id;
```

3.通过实时计算Flink写入数据

Blink 3.6.0版本开始支持通过Blink connector将数据写入云原生数据仓库AnalyticDB PostgreSQL版，本文将为您介绍使用的必要条件、操作流程、字段映射和参数支持。

前提条件

实时计算集群和AnalyticDB PostgreSQL版实例位于同一VPC下，且AnalyticDB PostgreSQL版实例的白名单规则允许Blink集群网段访问。

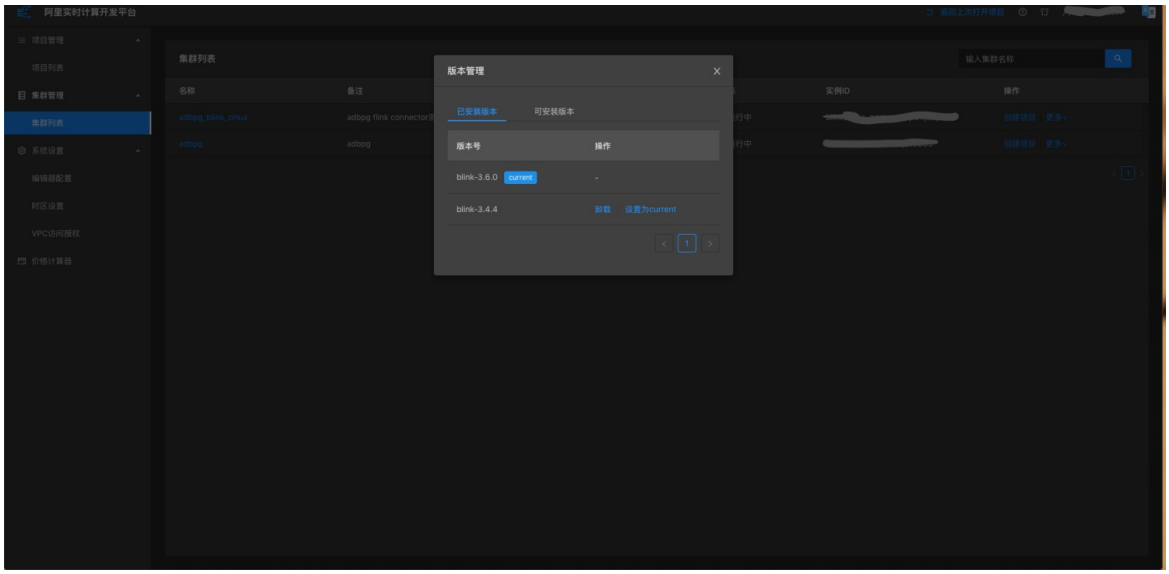
创建实时计算集群

实时计算集群为3.6.0及以上版本，可按以下步骤创建。

1. 开通阿里云实时计算服务和项目，请参见开通服务和创建项目。

   > **说明** 开通的实时计算集群与目标AnalyticDB PostgreSQL版集群必须在同一VPC下。

2. 确认并安装实时计算集群3.6.0及以上版本，请参见管理独享集群Blink版本。



配置AnalyticDB PostgreSQL 6.0版实例

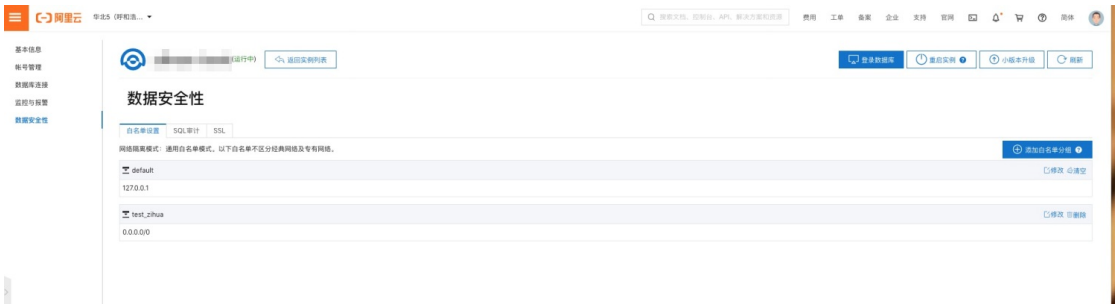
1. 创建实例。

   > **说明** 开通的AnalyticDB PostgreSQL版实例与实时计算集群必须在同一VPC下。

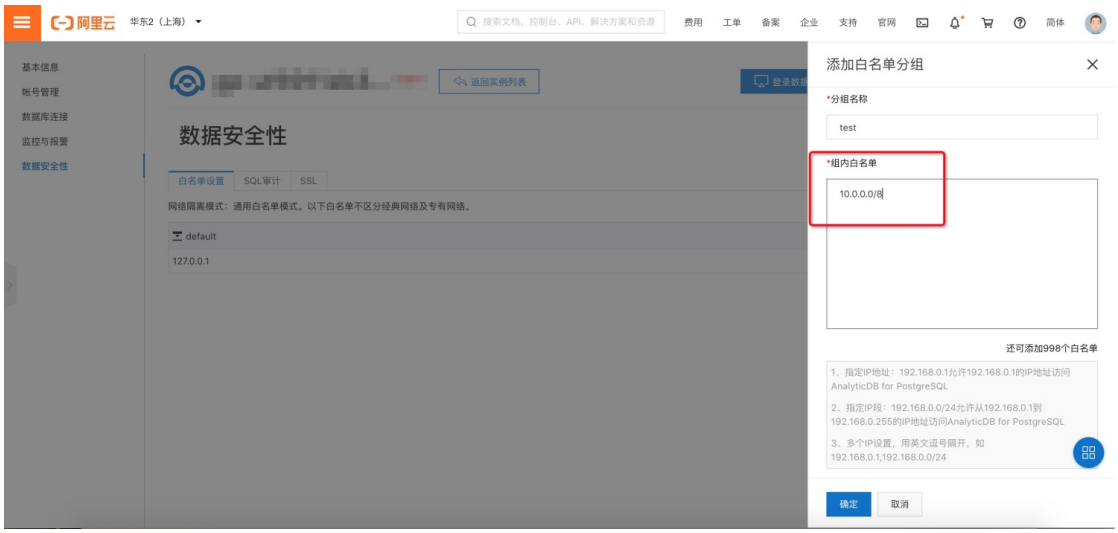
2. 设置AnalyticDB PostgreSQL版实例白名单。

i. 在VPC控制台找到对应网段的IP地址。

ii. 在ADB PG版控制台单击目标AnalyticDB PostgreSQL版实例ID，在实例详情页面，单击**数据安全性 > 添加白名单分组**。



iii. 将对应的VPC网段添加进AnalyticDB PostgreSQL版实例白名单，单击**确定**。



3. 创建AnalyticDB PostgreSQL版目标表。

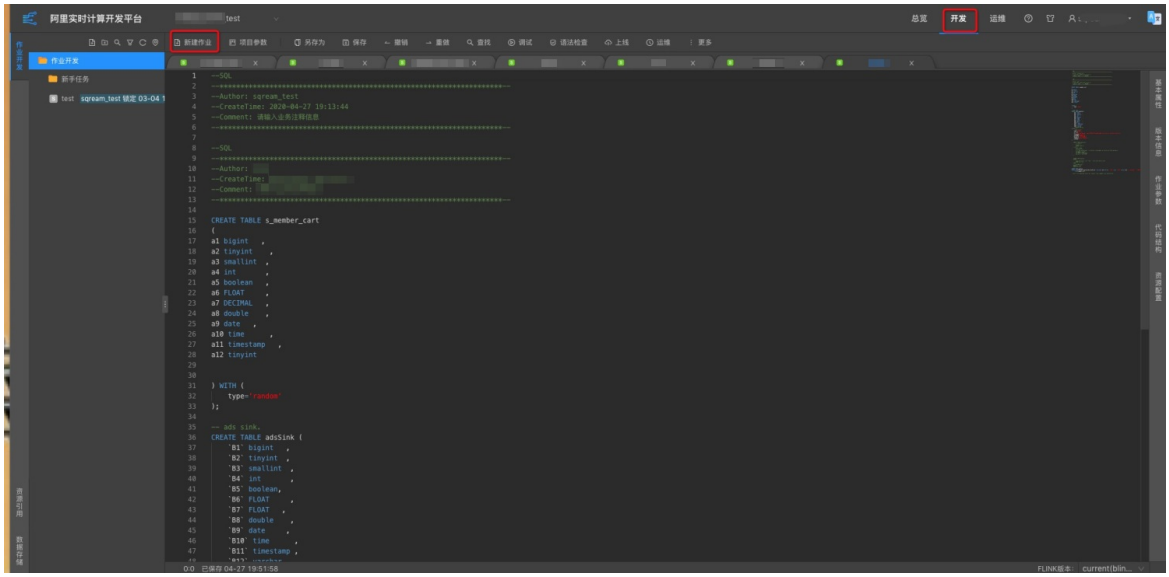
```
create table test15(
b1 bigint,
b2 smallint,
b3 smallint,
b4 int,
b5 boolean,
b6 real,
b7 double precision,
b8 double precision,
b9 date,
b10 time with time zone,
b11 timestamp with time zone,
b12 text,
b15 json
);
```

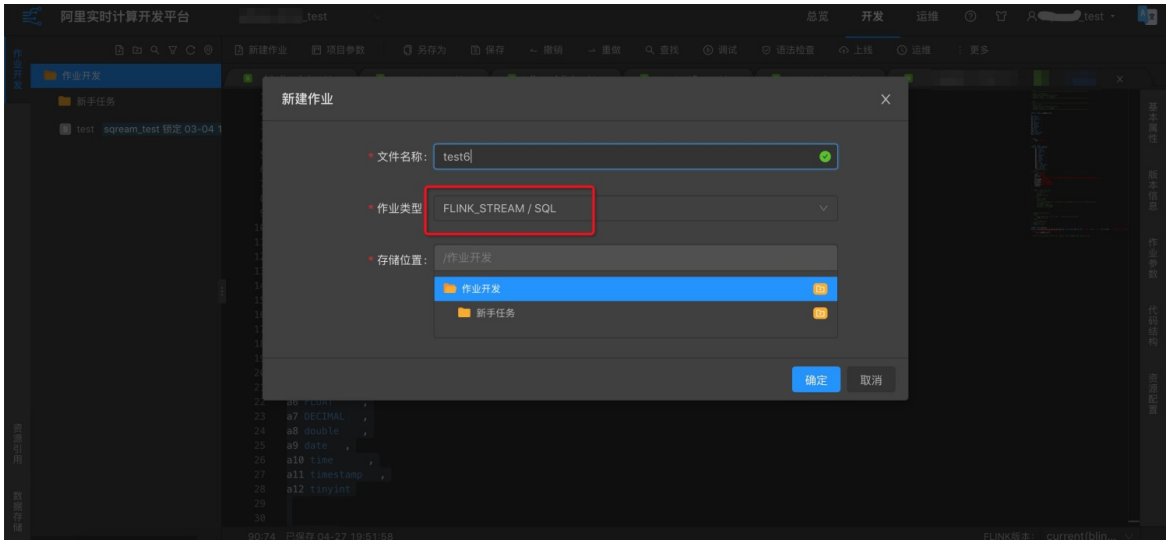
创建数据写入任务

为了方便介绍，本节的数据源采用随机数据源（random），实际使用中可以根据实际情况创建数据源。

1. 在实时计算控制台，单击**项目管理 > 项目列表**，单击项目名进入目标项目。

2. 在**开发**页面，单击**新建作业**，创建数据写入的Flink SQL作业。





写入AnalyticDB PostgreSQL版的作业举例。

```
--SQL
--********************************************************************--
--Author: sqream_test
--CreateTime: 2020-04-27 19:13:44
--********************************************************************--
CREATE TABLE s_member_cart
(
a1 bigint  ,
a2 tinyint  ,
a3 smallint ,
a4 int  ,
a5 boolean  ,
a6 FLOAT  ,
a7 DECIMAL  ,
a8 double  ,
a9 date  ,
a10 time  ,
a11 timestamp  ,
```

```
a12 tinyint
) WITH (
  type='random'
);
-- ads sink.
CREATE TABLE adsSink (
  `B1` bigint ,
  `B2` tinyint ,
  `B3` smallint ,
  `B4` int ,
  `B5` boolean,
  `B6` FLOAT ,
  `B7` FLOAT ,
  `B8` double ,
  `B9` date ,
  `B10` time ,
  `B11` timestamp ,
  `B12` varchar,
  `B15` varchar
  --PRIMARY KEY(b1)
) with (
  --type='print'
  type='adbpg',
  version='1.1',
  url='jdbc:postgresql://gp-xxxx:3432/testblink',
  tableName='test',
  userName='xxxx',
  password='xxxxxx',
  timeZone='Asia/Shanghai',
  useCopy='0'
);
INSERT INTO adsSink
SELECT a1,a2,a3,a4,a5,a6,a6,a8,a9,a10,a11, case when a12 >0 then 'value1' else 'value2' end as b12,'{ "custo
mer": "value", "items": {"product": "Beer","qty": 6}}'
  from s_member_cart;
--insert into adsSink2 select a2, sum(a4) from s_member_cart group by a2;
```

参数说明

参数	参数说明	是否必填	备注
type	源表类型	是	固定值：adbpg。

参数	参数说明	是否必填	备注
url	JDBC连接地址	是	AnalyticDB PostgreSQL版的JDBC连接地址 。格式为'jdbc:postgresql://<yourNetworkAddress>:<PortId>/<yourDatabaseName>'，其中yourNetworkAddress：内网地址。PortId：连接端口。yourDatabaseName：连接的数据库名称。示例：url='jdbc:postgresql://gp-xxxxxx.gpdb.cn-chengdu.rds.aliyuncs.com:3432/postgres'
tableName	表名	是	无。
username	账号	是	无。
password	密码	是	无。
maxRetryTimes	写入重试次数	否	默认为3。
useCopy	是否采用copy API写入数据	否	参数取值如下 • 1：采用copy API方式写入数据。 • 0：采用其他方式写入数据，例如BATCH INSERT或BATCH UPSERT。 Blink 3.6.0 版本默认为0，3.6.4及以上版本默认值为1；当取值为0时，会根据writeMode字段选择数据写入方式。
batchSize	一次批量写入的条数	否	默认值为5000。
exceptionMode	数据写入过程中出现异常时的处理策略	否	支持以下两种处理策略： • ignore（默认值）：忽略出现异常时写入的数据。 • strict：数据写入异常时，Failover报错。

参数	参数说明	是否必填	备注
conflictMode	当出现主键冲突或者唯一索引冲突时的处理策略	否	支持以下三种处理策略： • ignore （默认值）：忽略主键冲突，保留之前的数据。 • strict：主键冲突时，Failover报错。 • update：主键冲突时，更新新到的数据。 • upsert：主键冲突时，采用upsert方式写入数据。
targetSchema	Schema名称	否	默认值为public。
writeMode	在useCopy字段基础上，更细分的写入方式	否	Blink 3.6.4 以后版本开始支持，在useCopy字段为0的场景下，可以设定writeMode字段采用其他写入方式，参数取值如下： • 0：采用BATCH INSERT方式写入数据。 • 1（默认值）：采用COPY API写入数据。 • 2：采用BATCH UPSERT方式写入数据。upsert含义见使用INSERT ON CONFLICT覆盖写入数据。

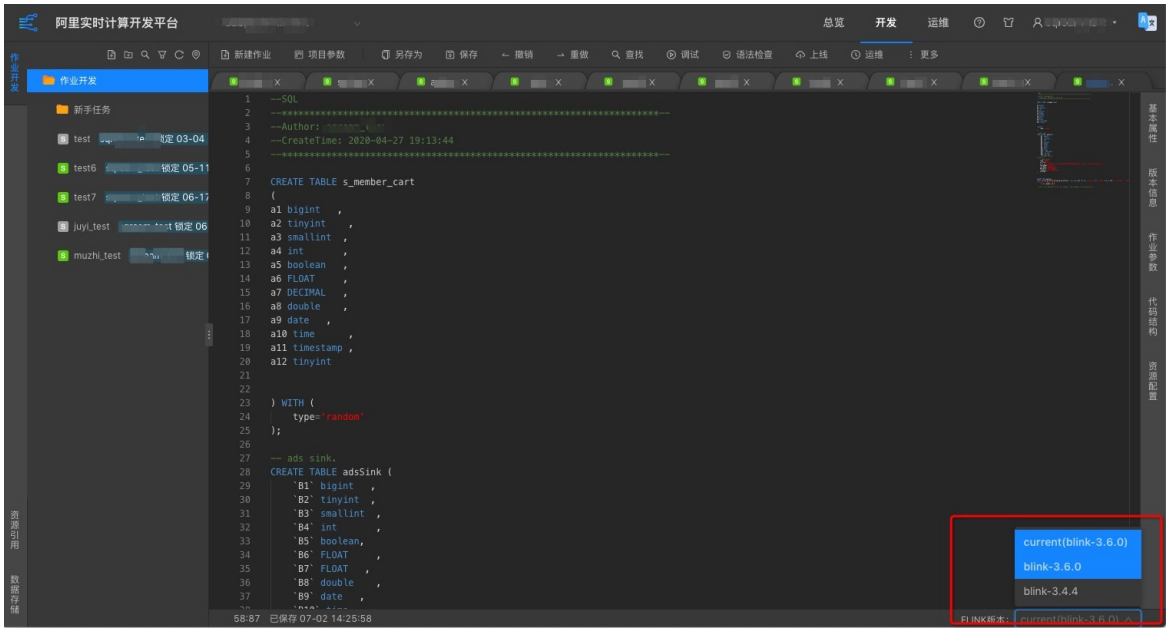
类型映射

实时计算字段类型	AnalyticDB PostgreSQL版字段类型
BOOLEAN	BOOLEAN
TINYINT	SAMLLINT
SAMLLINT	SAMLLINT
INT	INT
BIGINT	BIGINT
DOUBLE	DOUBLE PRECISION
VARCHAR	TEXT
DATETIME	TIMESTAMP

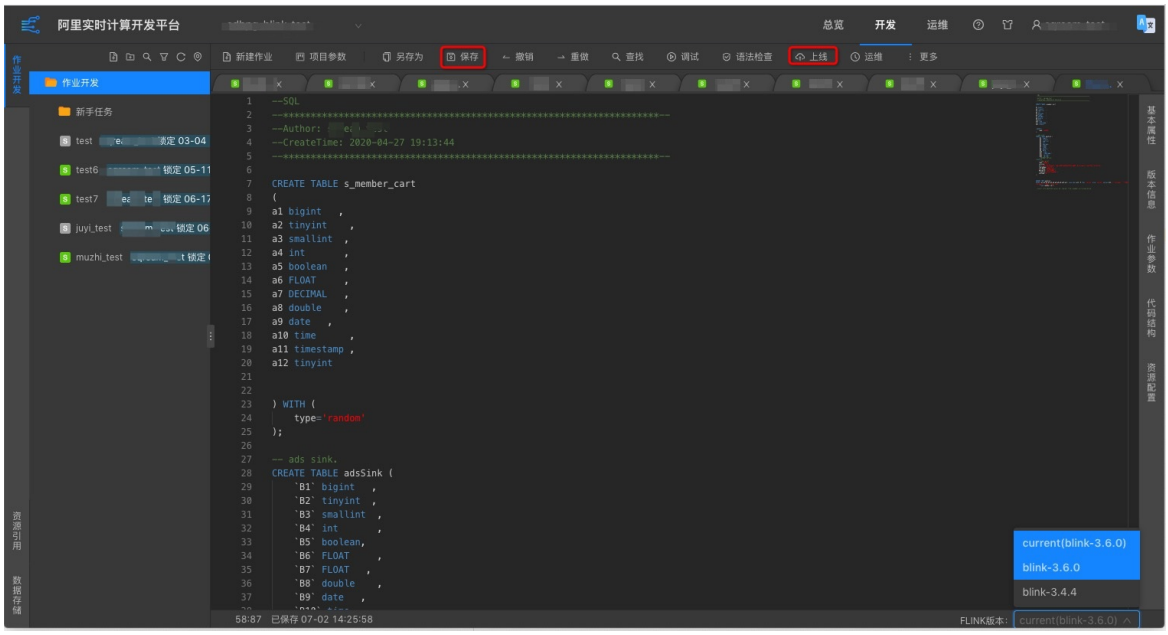
实时计算字段类型	AnalyticDB PostgreSQL版字段类型
DATE	DATE
FLOAT	REAL
DECIMAL	DOUBLE PRECISION
TIME	TIME
TIMESTAMP	TIMESTAMP

启动导入任务

1. 在开发作业页面的右下角确认当前作业版本为3.6.0及以上，如果不符请单击切换版本。



2. 完成作业开发后，依次单击**保存**、**上线**，即可上线该任务。



3. 单击**运维**，在运维页面单击目标项目操作栏中的**启动**即可开始导入。

连接对应AnalyticDB PostgreSQL版实例，发现数据已经写入了目标表。

```
testblink=> select * from test15;
          b1          |  b2  |   b3   |     b4      | b5 |     b6      |          b7          |          b8          |     b9
|     b10     |          b11           |  b12   |                  b15
----------------------+------+--------+-------------+----+-------------+----------------------+----------------------+------------
+-------------+------------------------+--------+-------------------------------------------------------------------------------
  6264464222845859665 |  -16 |  13112 |   940280953 | t  |  0.70453215 |  0.70453215000000002 |    0.5764313152073578 | 2015-05-01
| 00:16:25+08 | 2027-12-01 13:01:52+08 | value1 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
 -2297050138502582321 |  -21 |  27330 | -1855667185 | t  |  0.13331461 |           0.13331461 |  0.42460516838379192 | 2062-04-29
| 23:44:46+08 | 2060-02-29 01:21:05+08 | value2 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
 -4332416042448185437 | -101 | -23569 | -1813431851 | t  |  0.40770698 |           0.40770698 | 0.047220570407624507 | 2036-07-07
| 17:55:16+08 | 2035-09-12 10:59:09+08 | value1 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
 -3763735569687107593 |   11 |   7349 |    74485118 | f  |  0.89603788 |  0.89603790000000005 |    0.3702610547904267 | 2045-06-24
| 12:41:36+08 | 2004-07-12 20:08:20+08 | value1 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
 -2311750608684684293 |   45 |  17154 |   471902222 | f  | 0.013693213 | 0.013693212999999999 |  0.33750752744605739 | 2001-06-05
| 03:39:22+08 | 1979-12-21 18:33:14+08 | value2 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
  5597031460895688180 |  -45 |  24719 |   166633699 | f  |  0.89617109 |            0.8961711 |  0.33409636839673418 | 2021-06-28
| 22:41:10+08 | 2029-06-25 14:43:08+08 | value2 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
 -3697996724827757251 |  -22 | -32228 |   -55387262 | t  |  0.20284826 |           0.20284826 |  0.81653309468810564 | 1995-05-05
| 22:46:34+08 | 2033-11-06 10:06:11+08 | value2 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
 -7421611409470418201 |   59 | -12584 |  -265637833 | f  |  0.16507059 |  0.16507060000000001 |  0.77453945729613782 | 1974-02-04
| 18:03:05+08 | 2063-12-19 23:02:12+08 | value1 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
   -1264163298460163 | -122 |  28174 | -1744646274 | f  |  0.23956299 |           0.23956299 |  0.96640856406009457 | 2030-06-21
| 01:04:54+08 | 2058-09-05 22:24:05+08 | value2 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
  7592404831789906703 |  -88 |  30752 |  -574653828 | f  | 0.084314883 | 0.084314879999999995 |  0.86367212153332562 | 1997-12-31
| 19:04:36+08 | 2051-10-05 21:01:14+08 | value2 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
 -5527033960347520935 |  -48 |  -7995 | -1152087733 | f  |  0.70276994 |  0.70276994000000004 |  0.37170223227173615 | 1993-12-11
| 11:41:52+08 | 2062-07-14 04:46:34+08 | value2 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
  4475929972399550778 |   -5 |  -5408 | -1494978950 | f  | 0.029378235 | 0.029378234999999999 |  0.30360464482486804 | 1978-02-07
| 14:42:28+08 | 2050-06-23 23:22:38+08 | value1 | { "customer": "value", "items": {"product": "Beer","qty": 6}}
(12 rows)
```

版本变更记录

Blink 3.6.4版本：

- 默认写入方式由BATCH INSERT变为BATCH COPY，以提高写入性能。
- 增加writeMode字段
  - 在3.6.4版本以后，如果不设置useCopy字段为1，则writeMode字段无论为何值均采用BATCH COPY方式写入。
  - 例如：采用BATCH INSERT方式写入，需要设定useCopy=0, writeMode=0; 采用BATCH UPSERT方式写入，需要设置useCopy=0,writeMode=2。
  - 在以后的迭代中，会逐步放弃useCopy字段，请尽量采用writeMode字段配置写入方式。
- conflictMode字段增加upsert取值，通过insert on conflict的方式处理主键冲突。

blink 3.6.0 版本：

blink 3.6.0版本开始支持6.0版本AnalyticDB PostgreSQL版结果表。

4.ETL工具支持概览

支持的ETL工具见下，可以同时参见更详细的同步方案列表数据迁移及同步方案综述：

- 数据传输服务（DTS）：阿里云提供的实时数据同步服务，可以将其他数据源（RDS MySQL，ECS自建MySQL，PolarDB等）实时同步数据到AnalyticDB PostgreSQL版，构建实时数据仓库解决方案。
- 阿里云的数据集成服务（Data Integration）：阿里云提供的ETL工具。在数据集成服务中，将AnalyticDB PostgreSQL版配置为一个PostgreSQL数据库，即可实现其他数据源（RDS、MaxCompute、TableStore等）到AnalyticDB PostgreSQL版的数据同步。
  - 您可以直接从其他数据源读取数据，写入到AnalyticDB PostgreSQL版中。
  - 如果数据量较大，需要并发导入，则建议您先通过数据集成服务把数据从其他数据源导入到OSS，再通过OSS外部表导入AnalyticDB PostgreSQL版。
- Pentaho Kettle 数据集成软件：开源的ETL工具。
  - 支持将数据先通过Kettle导入到本地磁盘，再通过COPY或OSS导入到AnalyticDB PostgreSQL版。
  - 也支持将OSS存储挂载为本地虚拟磁盘，通过Kettle导入到此磁盘，最后通过AnalyticDB PostgreSQL版的OSS外部表导入到AnalyticDB PostgreSQL版中。
- Informatica软件：商业化的ETL工具。
- dbsync：阿里云提供的开源数据库同步工具。
  - 支持从MySQL、PostgreSQL并发同步数据到AnalyticDB PostgreSQL版。
  - 支持简单的数据转换。
  - 支持通过解析Binlog，准实时地从MySQL同步数据到AnalyticDB PostgreSQL版。
- 其他支持Greenplum的ETL工具。

5.Dataworks数据集成

数据集成（Data Integration）是阿里巴巴集团提供的数据同步平台。该平台具备可跨异构数据存储系统、可靠、安全、低成本、可弹性扩展等特点，可为20多种数据源提供不同网络环境下的离线（全量/增量）数据进出通道。

关于数据集成的更多信息，请参见数据集成（Data Integration）和支持的数据源与读写插件。

应用场景

- AnalyticDB PostgreSQL版可以通过数据集成的同步任务将数据同步到到其他的数据源中（AnalyticDB PostgreSQL版数据导出），并对数据进行相应的处理。
- 可以通过数据集成的同步任务将处理好的其他数据源数据同步到AnalyticDB PostgreSQL版（AnalyticDB PostgreSQL版数据导入）。

无论是哪种应用场景，都可以通过DataWorks的数据集成功能完成数据的同步过程，详细的操作步骤（包括创建数据集成任务、数据源配置、作业配置、白名单配置等），请参考DataWorks文档中的使用指南-->数据集成一栏。文章中余下部分会介绍AnalyticDB PostgreSQL版的数据导入导出操作详细步骤。

准备工作

数据集成任务准备

1. 开通准备阿里云账号
2. 开通MaxCompute，自动产生一个默认的MaxCompute数据源，并使用主账号登录DataWorks
3. 创建工作空间。您可在工作空间中协作完成工作流，共同维护数据和任务等，因此使用DataWorks前需要先创建工作空间。

> **说明** 如果您想通过子账号创建数据集成任务，可以赋予其相应的权限。详情请参见准备RAM用户

AnalyticDB PostgreSQL版准备：

1. 进行数据导入操作前，请通过PostgreSQL客户端创建好AnalyticDB PostgreSQL版中需要迁入数据的目标数据库和表。
2. 对于数据导出，请登录AnalyticDB PostgreSQL版的管理控制台进行IP**白名单设置**，详情请参见 添加白名单

数据导入

源端的数据源需要在DataWorks管理控制台进行添加，数据源添加的详细步骤请参考配置AnalyticDB for PostgreSQL数据源

配置同步任务：

配置好数据源后，就可以配置同步任务，完成数据源数据到AnalyticDB PostgreSQL版的数据导入。配置同步任务有两种模式：**向导模式**和**脚本模式**。

- 向导模式。通过向导模式配置数据集成任务，需要依次完成以下几步：
  i. 新建数据同步节点。
  ii. 选择数据来源。
  iii. 选择数据去向（这里的数据去向一定是AnalyticDB PostgreSQL版）。
  iv. 配置字段的映射关系。
  v. 配置作业速率上限、脏数据检查规则等信息。
  vi. 配置调度属性。

  > **说明** 具体操作步骤请参考DataWorks通过向导模式配置任务

- 脚本模式。通过脚本模式配置数据集成任务，需要依次完成以下几步：
  i. 新建数据同步节点。
  ii. 导入模板。
  iii. 配置同步任务的读取端。
  iv. 配置同步任务的写入端（这里写入端一定是AnalyticDB PostgreSQL版）。
  v. 配置字段的映射关系。
  vi. 配置作业速率上限、脏数据检查规则等信息。
  vii. 配置调度属性。

  > **说明** 具体操作步骤请参考DataWorks通过脚本模式配置任务

数据导出

数据导出的步骤和数据导入的步骤一样，区别是在数据导出中，数据源配置需要配置为AnalyticDB PostgreSQL版（参见配置AnalyticDB for PostgreSQL数据源），而目的端可以配置为其他的数据源类型。

参考信息

更多数据集成详细信息请参考DataWorks文档

6.使用\COPY命令导入本地数据

本文介绍如何通过\COPY命令，将本地的文本文件数据导入云原生数据仓库AnalyticDB PostgreSQL版数据库。

注意事项

由于\COPY命令需要通过Master节点进行串行数据写入处理，因此无法实现并行写入大批量数据。如果要进行大量数据的并行写入，请使用基于OSS的数据导入方式。通过OSS导入数据，请参见使用OSS外表高速导入OSS数据。

/COPY与COPY命令的区别

- 相同点：

  /COPY与COPY命令语法上没有任何区别。更多关于两种命令的使用方法，请参见COPY。

- 不同点：
  - /COPY为psql的操作指令；COPY为数据库指令。
  - /COPY命令支持操作FILE、STDIN和STDOUT文件；COPY命令仅支持操作STDIN和STDOUT文件，不支持操作FILE文件。

    > **说明** COPY命令需要SUPERUSER权限才支持操作FILE文件，目前AnalyticDB PostgreSQL不提供SUPERUSER权限。

语法

\COPY导入数据的语法如下：

```
\COPY table [(column [, ...])] FROM {'file' | STDIN}
[ [WITH]
[OIDS]
[HEADER]
[DELIMITER [ AS ] 'delimiter']
[NULL [ AS ] 'null string']
[ESCAPE [ AS ] 'escape' | 'OFF']
[NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
[CSV [QUOTE [ AS ] 'quote']
[FORCE NOT NULL column [, ...]]
[FILL MISSING FIELDS]
[[LOG ERRORS [INTO error_table] [KEEP]
SEGMENT REJECT LIMIT count [ROWS | PERCENT] ]
```

> **说明** AnalyticDB PostgreSQL支持使用JDBC执行COPY语句，JDBC中封装了CopyIn方法，具体信息，请参见Interface CopyIn。

示例

```
\COPY test1 FROM '/path/to/localfile';
```

相关文档

使用\COPY命令导出数据到本地

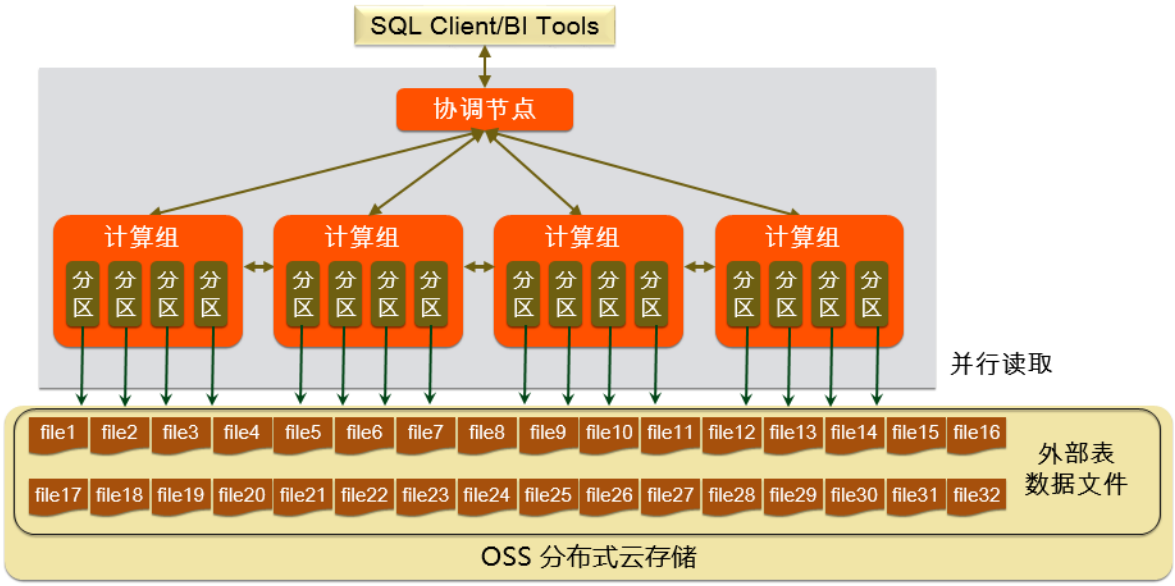
7.使用OSS外表高速导入OSS数据

云原生数据仓库AnalyticDB PostgreSQL版支持通过OSS外部表（gpossext功能），将数据并行从阿里云对象存储OSS导入到AnalyticDB PostgreSQL。

功能介绍

目前gpossext支持读写TEXT、CSV格式的文件以及GZIP压缩格式的TEXT、CSV文件。

gpossext架构图如下。



TEXT和CSV格式说明

下列几个参数可以在外表DDL参数中指定，用于规定读写OSS的文件格式：

- TEXT和CSV行分割符号是 `\n` ，也就是换行符。
- DELIMITER用于定义列的分割符：
  - 当用户数据中包括DELIMITER时，则需要和QUOTE参数一同使用。
  - 推荐的列分割符有 `,` 、 `\t` 、 `|` 或一些不常出现的字符。
- QUOTE用于包裹有特殊字符的用户数据（以列为单位）：
  - 包含有特殊字符的字符串会被QUOTE包裹，用于区分用户数据和控制字符。
  - 如果不必要，例如整数，基于优化效率的考虑，不必使用QUOTE包裹数据。
  - QUOTE不能和DELIMITER相同，默认QUOTE是双引号。
  - 当用户数据中包含了QUOTE字符，则需要使用转义字符ESCAPE加以区分。
- ESCAPE用于特殊字符转义：
  - 转义字符出现在需要转义的特殊字符前，表示它不是一个特殊字符。
  - ESCAPE默认和QUOTE相同，为双引号 `""` 。
  - 也支持设置成 `\` （MySQL默认的转义字符）或别的字符。

典型的TEXT和CSV默认控制字符

控制字符和格式	TEXT	CSV
DELIMITER（列分割符）	\t（Tab）	，（Comma）
QUOTE（摘引）	" （Double-Quote）	"（Double-Quote）
ESCAPE（转义）	（不适用）	与QUOTE相同
NULL（空值）	\N（Backslash-N）	（无引号的空字符串）

说明 所有的控制字符都必须是单字节字符。

注意事项

- 创建和使用外部表的语法，除了location相关参数，其余参数和Greenplum的使用方式相同。
- 数据导入导出的性能和AnalyticDB PostgreSQL的资源（CPU、I/O、内存、网络等）有关，也和OSS相关。为了获取最优的导入导出性能，建议在创建表时，使用列式存储加压缩功能。例如，指定子句 “WITH (APPENDONLY=true, ORIENTATION=column, COMPRESSTYPE=zlib, COMPRESSLEVEL=5, BLOCKSIZE=1048576)，详细信息，请参见 Greenplum Database 表创建语法官方文档。
- 为了保证数据导入导出的性能，请保证OSS与AnalyticDB PostgreSQL在同一地域下。关于OSS Endpoint的相关信息，请参见 OSS endpoint 信息。

操作步骤

1. 创建OSS外部表插件。

   使用OSS外部表时，需要在AnalyticDB PostgreSQL中先创建OSS外部表插件（每个库中均需要单独创建）。

   - 创建命令如下：

     ```
     CREATE EXTENSION IF NOT EXISTS oss_ext;
     ```

   - 删除命令如下：

     ```
     DROP EXTENSION IF EXISTS oss_ext;
     ```

2. 将待导入AnalyticDB PostgreSQL的数据均匀分散存储在多个OSS文件中。操作方式，请参见大文件切分。

   说明 AnalyticDB PostgreSQL的每个数据分区（Segment节点）将按轮询方式并行对OSS上的数据文件进行读取，文件的数目建议为数据节点数（Segment个数）的整数倍，从而提升读取效率。

3. 在AnalyticDB PostgreSQL中，创建READABLE外部表。

   创建OSS外部表语法如下。

```
CREATE [READABLE] EXTERNAL TABLE tablename
( columnname datatype [, ...] | LIKE othertable )
LOCATION ('ossprotocol')
FORMAT 'TEXT'
        [( [HEADER]
          [DELIMITER [AS] 'delimiter' | 'OFF']
          [NULL [AS] 'null string']
          [ESCAPE [AS] 'escape' | 'OFF']
          [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
          [FILL MISSING FIELDS] )]
        | 'CSV'
        [( [HEADER]
          [QUOTE [AS] 'quote']
          [DELIMITER [AS] 'delimiter']
          [NULL [AS] 'null string']
          [FORCE NOT NULL column [, ...]]
          [ESCAPE [AS] 'escape']
          [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
          [FILL MISSING FIELDS] )]
[ ENCODING 'encoding' ]
[ [LOG ERRORS [INTO error_table]] SEGMENT REJECT LIMIT count
     [ROWS | PERCENT] ]
ossprotocol:
  oss://oss_endpoint [prefix=prefix_name|dir=[folder/[folder/]...]/file_name|filepath=[folder/[folder/]...]/
file_name]
   id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
```

参数说明如下。

参数	说明
FORMAT	支持文件格式，例如TEXT、CSV。
ENCODING	文件中数据的编码格式，例如UTF8。

参数	说明

LOG ERRORS	指定该子句可以忽略掉导入中出错的数据，将这些数据写入error_table，并可以使用count参数指定报错的阈值。
	说明
	○ 通过 `LOG ERRORS` 将错误行信息记录到内部关联文件。
	`LOG ERRORS SEGMENT REJECT LIMIT 5;`
	○ 通过函数 `gp_read_error_log('external_table_name')` 可以读取错误行信息。
	`SELECT * FROM gp_read_error_log('external_table_name');`
	○ 内部文件随外表删除而删除，也可以通过函数 `gp_truncate_error_log('external_table_name')` 删除。
	`SELECT gp_truncate_error_log('external_table_name');`

oss://oss_endpoint	协议和Endpoint，格式为 **协议名://oss_endpoint** ，其中协议名为oss，oss_endpoint为OSS对应区域的域名。示例如下： **oss://oss-cn-hangzhou.aliyuncs.com** **注意** 如果是从阿里云的主机访问数据库，应该使用内网域名（即带有 `internal` 的域名），避免产生公网流量。
id	阿里云账号的AccessKey ID。获取AccessKey操作，请参见获取AccessKey。
key	阿里云账号的AccessKey Secret。获取AccessKey操作，请参见获取AccessKey。
bucket	指定数据文件所在的Bucket，需要在OSS上预先创建。

参数	说明
prefix	指定数据文件对应路径名的前缀，不支持正则表达式，仅是匹配前缀。 **说明** 与dir、filepath互斥，三者只能设置其中一个。 READABLE外部表会在导入数据时将含有这一前缀的所有OSS文件都会被导入。 ○ 如果指定prefix=test/filename，以下文件都会被导入： ■ test/filename ■ test/filenamexxx ■ test/filename/aa ■ test/filenameyyy/aa ■ test/filenameyyy/bb/aa ○ 如果指定prefix=test/filename/，只有以下文件会被导入（上面列的其他文件不会被导入）： ■ test/filename/aa
dir	OSS中的虚拟文件夹路径。 **说明** 与prefix、filepath互斥，三者只能设置其中一个。 ○ 文件夹路径需要以 / 结尾，如 test/mydir/ 。 ○ 在导入数据时，使用此参数创建外部表，会导入指定虚拟目录下的所有文件，但不包括它子目录和子目录下的文件。与filepath不同，dir下的文件没有命名要求。
filepath	OSS中包含路径的文件名称。 **说明** ○ 与prefix、dir互斥，三者只能设置其中一个。 ○ 这个参数只能在创建READABLE外部表时指定，即仅支持在导入数据时使用。
compressiontype	导入文件的压缩格式。 ○ none（默认值）：导入的文件未压缩。 ○ gzip：导入的文件压缩格式为GZIP。 **说明** 目前仅支持GZIP压缩格式。
compressionlevel	设置写入OSS的文件的压缩等级，取值范围为1~9，默认值为6。示例如下： compressionlevel=6

参数	说明
oss_connect_timeout	设置连接超时。单位为秒，默认为10秒。
oss_dns_cache_timeout	设置DNS超时。单位为秒，默认为60秒。
oss_speed_limit	设置触发超时的最小速率。默认为1024字节，即1 KB。 需要与oss_speed_time参数配合使用。 **说明** 如果使用默认值且连续15秒的传输速率小于1 KB，会触发超时。具体信息，请参见OSS SDK 错误处理。
oss_speed_time	设置触发超时的最长时间。默认为15秒。 需要与oss_speed_limit参数配合使用。 **说明** 如果使用默认值且连续15秒的传输速率小于1 KB，会触发超时。具体信息，请参见OSS SDK 错误处理。
async	是否启用异步模式导入数据。 ○ 开启辅助线程从OSS导入数据，加速导入性能。 ○ 默认情况下异步模式为开启状态，如果需要关掉，可以使用参数 async = false 或 async = f 。 ○ 异步模式和普通模式比，会消耗更多的硬件资源。

4. 并行导入数据。

   在AnalyticDB PostgreSQL数据库中执行如下命令，将OSS上的数据并行导入到AnalyticDB PostgreSQL数据库。

```
INSERT INTO <目标表> SELECT * FROM <外部表>
```

操作示例

本文以目标表example为例，介绍将OSS的数据通过外部表导入目标表example。

1. 创建OSS外部表插件。

   创建命令如下：

```
CREATE EXTENSION IF NOT EXISTS oss_ext;
```

2. 创建目标表，用于装载数据。

```
CREATE TABLE example
    (date text, time text, open float,
     high float, low float, volume int)
     DISTRIBUTED BY (date);
```

3. 创建OSS导入外部表。
   - 创建外部表时，使用prefix参数指定待导入数据所在的路径。示例如下：

```
CREATE READABLE EXTERNAL TABLE ossexample
    (date text, time text, open float, high float,
    low float, volume int)
    location('oss://oss-cn-hangzhou.aliyuncs.com
    prefix=osstest/example id=XXX
    key=XXX bucket=testbucket compressiontype=gzip')
    FORMAT 'csv' (QUOTE '"' DELIMITER E'\t')
    ENCODING 'utf8'
    LOG ERRORS SEGMENT REJECT LIMIT 5;
```

- 创建外部表时，使用dir参数指定待导入数据所在的路径。示例如下：

```
CREATE READABLE EXTERNAL TABLE ossexample
    (date text, time text, open float, high float,
    low float, volume int)
    location('oss://oss-cn-hangzhou.aliyuncs.com
    dir=osstest/ id=XXX
    key=XXX bucket=testbucket')
    FORMAT 'csv'
    LOG ERRORS SEGMENT REJECT LIMIT 5;
```

- 创建外部表时，使用filepath参数指定待导入数据所在的路径。示例如下：

```
CREATE READABLE EXTERNAL TABLE ossexample
    (date text, time text, open float, high float,
    low float, volume int)
    location('oss://oss-cn-hangzhou.aliyuncs.com
    filepath=osstest/example.csv id=XXX
    key=XXX bucket=testbucket')
    FORMAT 'csv'
    LOG ERRORS SEGMENT REJECT LIMIT 5;
```

4. 将数据并行从ossexample外部表导入example表中。

```
INSERT INTO example SELECT * FROM ossexample;
```

执行如下查询计划，可以看到每个Segment节点都会从OSS并行拉取数据，然后通过执行节点Redistribution Motion将数据HASH计算后分发给对应的Segment节点，接收数据的Segment节点通过INSERT执行节点进行入库。

```
EXPLAIN INSERT INTO example SELECT * FROM ossexample;
                            QUERY PLAN
-------------------------------------------------------------------------------------------------
 Insert (slice0; segments: 4)  (rows=250000 width=92)
   ->  Redistribute Motion 4:4  (slice1; segments: 4)  (cost=0.00..11000.00 rows=250000 width=92)
       Hash Key: ossexample.date
       ->  External Scan on ossexample  (cost=0.00..11000.00 rows=250000 width=92)
(4 rows)
```

SDK错误处理

当导入或导出操作出错时，错误日志可能会出现如下信息：

- code：出错请求的HTTP状态码。

- error_code：OSS的错误码。
- error_msg：OSS的错误信息。
- req_id：标识该次请求的UUID。当您无法解决问题时，可以凭req_id来请求OSS开发工程师的帮助。

具体信息，请参见OSS API 错误响应，超时相关的错误可以使用oss_ext相关参数处理。

参考文档

- OSS Endpoint 信息
- OSS帮助文档
- OSS SDK错误处理
- OSS API错误响应
- Greenplum Database外部表语法官方文档
- Greenplum Database表创建语法官方文档

8.通过外表在HDFS上读写数据

AnalyticDB PostgreSQL版支持在Hadoop集群上读写数据。本文主要介绍在AnalyticDB PostgreSQL版中使用gphdfs协议向HDFS读写数据的步骤。

> **说明** 如需使用该功能，请提交工单联系技术支持开通。

创建HDFS测试文件

登录到HDFS，创建相应测试目录和文件，操作示例如下：

```
root@namenode:/# hadoop fs -mkdir /test #在根目录下创建一个test文件夹
root@namenode:/# echo "1 abc" > data.txt #创建一个测试数据文件
root@namenode:/# hadoop fs -put local.txt /test #上传测试数据文件到hdfs中
root@namenode:/# hadoop fs -cat /test/data.txt #查看文件内容
1 abc
```

创建HDFS读外表并查询数据

在外表创建语句中，指定HDFS集群的地址（使用gphdfs外表访问协议），以及关联的文件路径，文件格式和分隔符。更多关于创建外表的信息，请参见CREATE EXTERNAL TABLE。

```
CREATE READABLE EXTERNAL TABLE test (id int, name text)
LOCATION ('gphdfs://namenode_IP:port/test/data.txt')
FORMAT 'text' (delimiter ' ');
```

从外部表中读取数据：

```
SELECT * FROM test;
```

查询结果如下：

```
 id | name
----+------
  1 | abc
(1 row)
```

创建HDFS写外表并写入数据

在创建外部表的语句中，声明WRITABLE，表示可写外部表：

```
CREATE WRITABLE EXTERNAL TABLE test_write (id int, name text)
LOCATION('gphdfs://namenode_IP:port/test/data.txt')
FORMAT 'text' (delimiter ' ');
```

使用INSERT语句写入数据：

```
INSERT INTO test_write VALUES(2, 'def');
```

在HDFS集群上查看文件，确认数据已正确写入：

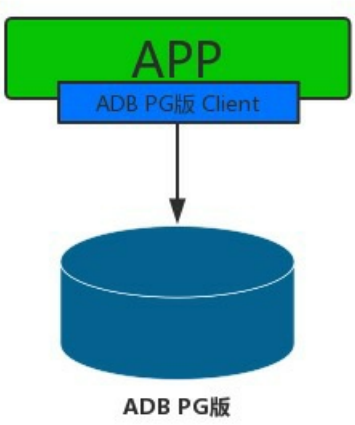
```
root@namenode:/# hadoop fs -cat /test/data.txt #查看文件内容
1 abc
2 def
```

9.基于Client SDK数据写入

AnalyticDB PostgreSQL版Client SDK旨在通过 API 方式提供高性能COPY数据到AnalyticDB PostgreSQL版的方式。

AnalyticDB PostgreSQL版Client SDK通过 API 形式旨在为用户提供高性能写入数据到AnalyticDB PostgreSQL版的方式，支持用户定制化开发或对接写入程序。通过 SDK 开发写入程序，可简化在AnalyticDB PostgreSQL版中写入数据的流程，无需担心连接池、缓存等问题，相比较直接COPY/INSERT写入，通过并行化等内部机制有几倍性能提升

> **说明** AnalyticDB PostgreSQL版Client SDK主要职责是将您传入的数据高效地写入，不负责原始数据的读取、处理等工作。



Maven repositories

您可以通过Maven管理配置新SDK的版本。Maven的配置信息如下：

```
<dependency>
 <groupId>com.alibaba.cloud.analyticdb</groupId>
 <artifactId>adb4pgclient</artifactId>
 <version>1.0.3</version>
</dependency>
```

> **说明**
>
> - AnalyticDB PostgreSQL版Client SDK本身依赖*druid(1.1.17)*、*postgresql(jdbc 42.2.5)*、*commons-lang3(3.4)*、*slf4j-api(1.7.24)*、*slf4j-log4j12(1.7.24)*。
> - 如果在使用过程中出现版本冲突，请检查这几个包的版本并解决冲突。

接口列表

DatabaseConfig类

接口名	描述
setHost(String adbHost)	需要连接的AnalyticDB PostgreSQL版的连接地址。
setPort(int port)	需要连接的AnalyticDB PostgreSQL版的端口，默认为3432。
setDatabase(String database)	需要连接的AnalyticDB PostgreSQL版数据库名称。
setUser(String username)	需要连接的AnalyticDB PostgreSQL版使用的用户名。
setPassword(String pwd)	设置连接的AnalyticDB PostgreSQL版使用的密码。
addTable(List<String> table, String schema)	需要写入的表名List，请按照表所属schema分别添加。该方法可调用多次，但在使用DatabaseConfig构造Adb4PGClient对象之后再调用不再生效。
setColumns(List<String> columns, String tableName, String schemaName)	需要插入表的字段名（若是全字段插入，`columnList.add("*")`即可，table列表中的所有表都需要设置字段名，不然检查不会通过
setInsertIgnore(boolean insertIgnore)	设置是否忽略发生主键冲突错误的数据行，要根据业务的使用场景进行判断，针对配置的所有表，默认为true
setEmptyAsNull(boolean emptyAsNull)	设置empty数据设置为null，默认false，针对配置的所有表
setParallelNumber(int parallelNumber)	设置写入ADB PG版时的并发线程数，默认4，针对配置的所有表，一般情况不建议修改
setLogger(Logger logger)	设置client中使用的logger对象，此处使用slf4j.Logger
setRetryTimes(int retryTimes)	设置commit时，写入ADB PG版出现异常时重试的次数，默认为3
setRetryIntervalTime(long retryIntervalTime)	设置重试间隔的时间，单位是ms，默认为 1000 ms
setCommitSize(long commitSize)	设置自动提交的数据量（单位Byte），默认为10MB，一般不建议设置

Row类

接口名称	描述
setColumn(int index, Object value)	设置Row字段列表的值，要求必须按照字段的顺序（此种方式，Row实例不可复用，每条数据必须单独的Row实例）
setColumnValues(List<Object> values)	直接将List格式数据行写入Row中
updateColumn(int index, Object value)	更新Row字段列表的值，注意更新的字段数据（此方法，Row实例可以复用，只需更新Row实例中的数据即可

Adb4pgClient 类

接口名称	描述
addRow(Row row, String tableName, String schemaName) / addRows(List<Row> rows, String tableName, String schemaName)	插入对应表的Row格式化的数据，即一条记录，数据会存储在SDK的缓冲区中，等待commit。如果数据量超过commitSize会在addRow/addRows的时候做一次自动commit，然后将最新的数据add进来；如果在自动commit失败的时候失败，调用方需要处理此异常，并且会在异常中得到失败的数据list
addMap(Map<String, String> dataMap,String tableName, String schemaName) / addMaps(List<Map<String, String>> dataMaps, String tableName, String schemaName)	对应于addRow，支持map格式数据的写入，如果数据量满了会在addMap/addMaps的时候做一次自动commit，然后将最新的数据add进来；如果在自动commit失败的时候失败，调用方需要处理此异常，并且会在异常中得到失败的数据list
commit()	将缓存的数据进行提交，写入ADB PG版中，若commit失败，会把执行错误的语句放在异常中抛出，调用方需要对此异常进行处理
TableInfo getTableInfo(String tableName, String schemaName)	获取对应table的结构信息
List<ColumnInfo> getColumnInfo(String tableName, String schemaName)	获取对应table的字段列表信息，字段类是ColumnInfo，可以通过columnInfo.isNullable()获取该字段是否能为null
stop()	实例使用完之后，stop释放内部线程池及资源，如果内存中有数据未commit，则会抛Exception，若需要强行stop，请使用forceStop()
forceStop()	强行释放内部线程池及资源，会丢失掉缓存在内存中未commit的数据，一般不推荐使用
Connection getConnection() throws SQLException	从client连接池获取ADB PG Connection连接，调用方可以使用获得的Connection做非copy操作，使用方式和jdbc的连接使用方式一致。 **说明** 使用结束后一定要释放掉相应的资源（如ResultSet、Statement、Connection）

ColumnInfo类

接口名称	描述
boolean isNullable()	判断该字段是否能为null

错误码名	错误码值	描述

错误码名	错误码值	描述
COMMIT_ERROR_DATA_LIST	101	commit中某些数据出现异常，会返回异常的数据。 说明 通过e.getErrData()即可获得异常数据List<String>，此错误码在addMap(s)、addRow(s)、commit操作的时候都可能会发生，因此在这些操作的时候需要单独处理此错误码的异常
COMMIT_ERROR_OTHER	102	commit中的其他异常
ADD_DATA_ERROR	103	add数据过程中出现的异常
CREATE_CONNECTION_ERROR	104	创建连接出现异常
CLOSE_CONNECTION_ERROR	105	关闭连接出现异常
CONFIG_ERROR	106	配置DatabaseConfig出现配置错误
STOP_ERROR	107	停止实例时的报错
OTHER	999	默认异常错误码

代码示例

```
public class Adb4pgClientUsage {
  public void demo() {
    DatabaseConfig databaseConfig = new DatabaseConfig();
    // Should set your database real host or url
    databaseConfig.setHost("100.100.100.100");
    // Should set your database real port
    databaseConfig.setPort(8888);
    // 连接数据库的用户名
    databaseConfig.setUser("your user name");
    // 连接数据库的密码
    databaseConfig.setPassword("your password");
   // 需要连接的database
    databaseConfig.setDatabase("your database name");
    // 设置需要写入的表名列表
    List<String> tables = new ArrayList<String>();
    tables.add("your table name 1");
    tables.add("your table name 2");
    // 不同schema下的表可分别addTable，但是一旦使用databseconfig 创造Client实例之后，table配置是不可修改的/
    // schema传入null, 则默认schema为public
    databaseConfig.addTable(tables, "table schema name");
    // 设置需要写入的表字段
    List<String> columns = new ArrayList<String>();
```

```
        columns.add("column1");
        columns.add("column2");
        // 如果是所有字段，字段列表使用 columns.add("*") 即可
        databaseConfig.setColumns(columns, "your table name 1", "table schema name");
        databaseConfig.setColumns(Collections.singletonList("*"),"your table name 2", "table schema name");
        // If the value of column is empty, set null
        databaseConfig.setEmptyAsNull(false);
        // 使用insert ignore into方式进行插入
        databaseConfig.setInsertIgnore(true);
        // commit时，写入ADB出现异常时重试的3次
        databaseConfig.setRetryTimes(3);
        // 重试间隔的时间为1s，单位是ms
        databaseConfig.setRetryIntervalTime(1000);
        // Initialize AdbClient，初始化实例之后，databaseConfig的配置信息不能再修改
        Adb4pgClient adbClient = new Adb4pgClient(databaseConfig);
        // 数据需要攒批，多次add，再commit，具体攒批数量见"注意事项"
        for (int i = 0; i < 10; i++) {
          // Add row(s) to buffer. One instance for one record
          Row row = new Row(columns.size());
          // Set column
          // the column index must be same as the sequence of columns
          // the column value can be any type, internally it will be formatted according to column type
          row.setColumn(0, i); // Number value
          row.setColumn(1, "string value"); // String value
          // 如果sql长度满了会在addRow或者addMap的时候会进行一次自动提交
          // 如果提交失败会返回AdbClientException异常，错误码为COMMIT_ERROR_DATA_LIST
          adbClient.addRow(row, "your table name 1", "table schema name");
        }
        Row row = new Row();
        row.setColumn(0, 10); // Number value
        row.setColumn(1, "2018-01-01 08:00:00"); // Date/Timestamp/Time value
        adbClient.addRow(row, "your table name 1", "table schema name");
        // Update column. Row实例可复用
        row.updateColumn(0, 11);
        row.updateColumn(1, "2018-01-02 08:00:00");
        adbClient.addRow(row, "your table name 1", "table schema name");
        // Add map(s) to buffer
        Map<String, String> rowMap = new HashMap<String, String>();
        rowMap.put("t1", "12");
        rowMap.put("t2", "string value");
        // 这边需要攒批的，最好多次add之后在进行commit
        adbClient.addMap(rowMap, "your table name 2", "table schema name");
        // Commit buffer to ADS
        // Buffer is cleaned after successfully commit to ADS
        try {
          adbClient.commit();
        } catch (Exception e) {
          // TODO: Handle exception here
        } finally {
          adbClient.stop();
        }
      }
    }
```

注意事项

- ADB PG 版Client SDK是非线程安全的，所以如果多线程调用的情况，需要每个线程维护自己的Client对象

  > **注意** 强烈不建议多线程共用SDK实例，除了线程安全问题外，容易让Client成为写入性能的瓶颈。

- 数据必须在调用commit成功后才能认为是写入ADB PG版成功的。
- 针对Client抛出的异常，调用方要根据错误码的意义自行判断如何处理，如果是数据写入有问题，可以重复提交或者记录下有问题的数据后跳过。
- 很多时候写入线程并不是越多越好，因为业务程序会涉及到攒数据的场景，对内存的消耗是比较明显的，所以业务调用方一定要多多关注应用程序的GC情况。
- 数据攒批数量不要太小，如果太小，攒批写入意义就不大了，条件允许的情况下可以add 10000条进行一次commit。
- DatabaseConfig配置在实例化client对象成功之后是不能再修改的，所有配置项必须在client对象初始化之前完成配置。
- Client SDK目的是对写入（INSERT）提供性能优化，对于其他SQL操作，可以通过getConnection()获得JDBC连接，通过标准JDBC接口进行处理

10.DTS数据迁移及同步方案列表

10.1. 从RDS SQL Server同步至云原生数据仓库AnalyticDB PostgreSQL

数据传输服务DTS（Data Transmission Service）支持将SQL Server同步至云原生数据仓库AnalyticDB PostgreSQL，帮助您轻松实现数据的流转，集中分析企业数据。

前提条件

- 创建RDS SQL Server实例，支持的版本，请参见同步方案概览。

  > 注意　目前支持源RDS SQL Server版本为2012、2016或2017版本。

- 创建云原生数据仓库AnalyticDB PostgreSQL实例。
- RDS SQL Server实例中待同步的表需具备主键。
- 云原生数据仓库AnalyticDB PostgreSQL实例中同步的目标表需具备主键或唯一索引。

注意事项

- DTS在执行全量数据迁移时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据迁移前评估源库和目标库的性能，同时建议您在业务低峰期执行数据迁移（例如源库和目标库的CPU负载在30%以下）。
- 为保障数据同步延迟显示的准确性，DTS会在源库中新增一张心跳表（名称为 `dts_log_heart_beat` ）。
- 此场景中，DTS支持初始化的结构为Schema、Table、View、Function和Procedure。

  > 警告　由于此场景属于异构数据库间的数据同步，数据类型无法一一对应，请谨慎评估数据类型的映射关系对业务的影响，详情请参见结构初始化涉及的数据类型映射关系。

- 不支持同步数据类型为TIMESTAMP、CURSOR、ROWVERSION、HIERACHYID、SQL_VARIANT、SPATIAL GEOMETRY、SPATIAL GEOGRAPHY、TABLE的数据。

支持同步的SQL操作

- DDL操作：ADD COLUMN
- DML操作：INSERT、UPDATE、DELETE

数据库账号的权限要求

数据库	所需权限	授权方法
RDS SQL Server实例	待同步数据库的所有者权限。	修改账号权限

数据库	所需权限	授权方法
云原生数据仓库AnalyticDB PostgreSQL实例	• LOGIN权限。 • 目标表的SELECT、CREATE、INSERT、UPDATE、DELETE权限。 • 目标库的CONNECT、CREATE权限。 • 目标Schema的CREATE权限。 • Copy权限（基于内存batch copy）。 **说明** 您也可以使用云原生数据仓库AnalyticDB PostgreSQL实例的初始账号。	用户权限管理

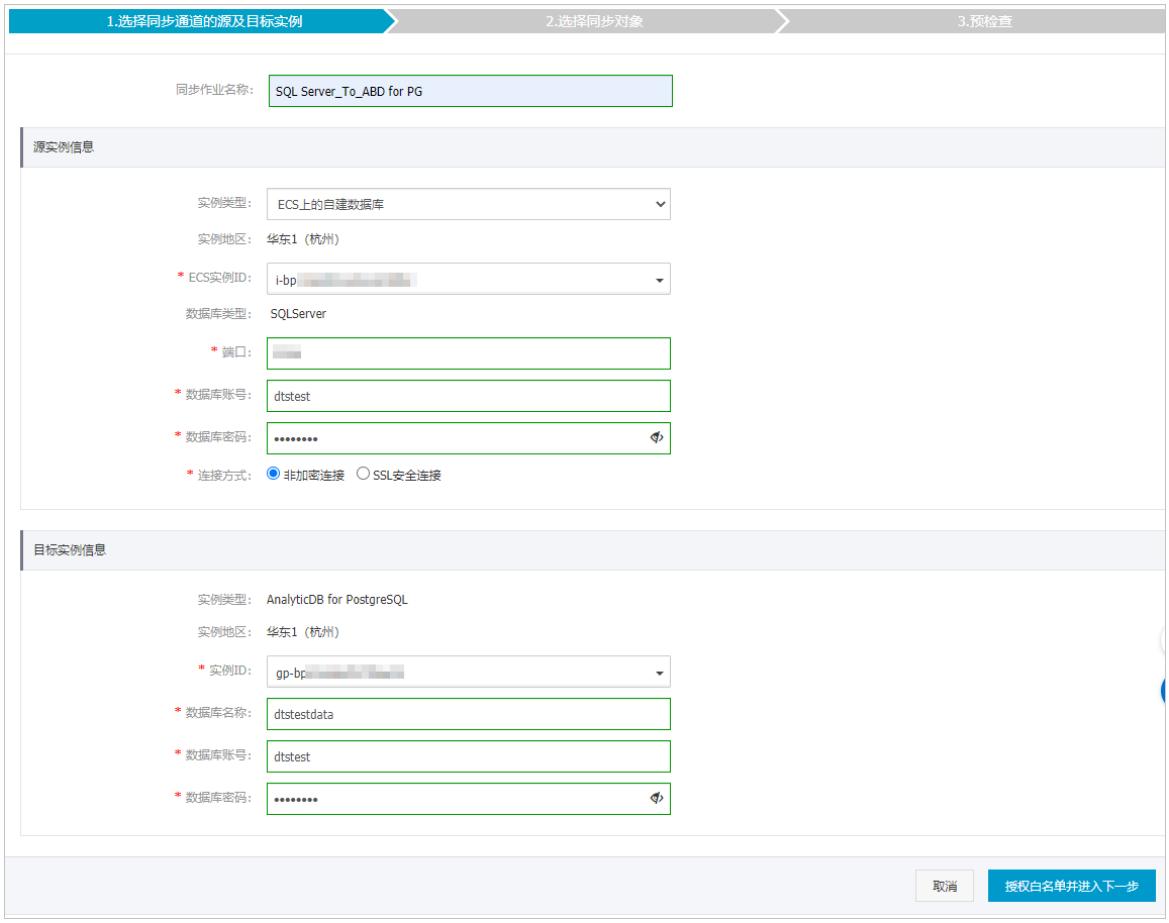
操作步骤

1. 购买数据同步作业，详情请参见购买流程。

   说明 购买时，选择源实例为SQLServer，目标实例为AnalyticDB for PostgreSQL，并选择同步拓扑为**单向同步**。

2. 登录数据传输控制台。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。

5. 定位至已购买的数据同步实例，单击**配置同步链路**。
6. 配置同步作业的源实例及目标实例信息。



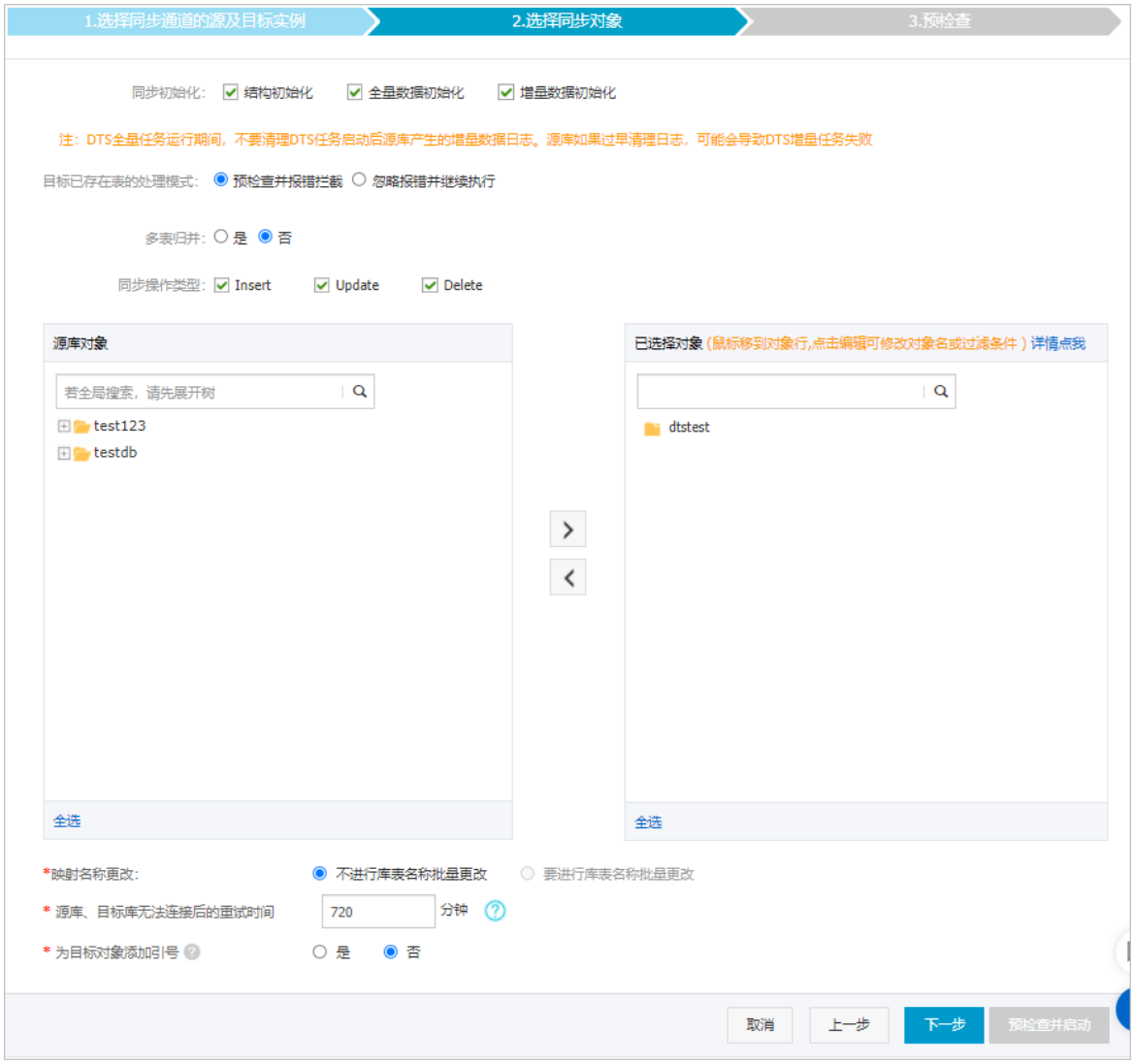
类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
源实例信息	实例类型	选择**RDS实例**。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	实例ID	选择源RDS SQL Server实例ID。
	数据库账号	填入RDS SQL Server的数据库账号。权限要求请参见数据库账号的权限要求 。
	数据库密码	填入该数据库账号的密码。
	连接方式	根据需求选择**非加密连接**或**SSL安全连接**。如果设置为**SSL安全连接**，您需要提前开启RDS实例的SSL加密功能，详情请参见设置SSL加密。
	实例类型	固定为**AnalyticDB for PostgreSQL**，无需设置。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	实例ID	选择目标云原生数据仓库AnalyticDB PostgreSQL实例ID。

目标实例信息	配置	说明
	数据库名称	填入同步目标表所属的数据库名称。
	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL的数据库账号。权限要求请参见数据库账号的权限要求 。
	数据库密码	填入该数据库账号对应的密码。

7. 单击页面右下角的**授权白名单并进入下一步**。

> **说明** 此步骤会将DTS服务器的IP地址自动添加到RDS SQL Server和云原生数据仓库AnalyticDB PostgreSQL的白名单中，用于保障DTS服务器能够正常连接源和目标实例。

8. 配置同步策略和同步对象。



配置	说明

配置	说明
同步初始化	默认选中**结构初始化**、**全量数据初始化**和**增量数据初始化**。预检查完成后，DTS会将源实例中待同步对象的结构和存量数据同步至目标在目标库，作为后续增量同步数据的基线数据。
目标已存在表的处理模式	○ **预检查并报错拦截**：检查目标数据库中是否有同名的表。如果目标数据库中没有同名的表，则通过该检查项目；如果目标数据库中有同名的表，则在预检查阶段提示错误，数据同步作业不会被启动。 **说明** 如果目标库中同名的表不能删除或重命名，您可以更改该表在目标库中的名称，详情请参见设置同步对象在目标实例中的名称。 ○ **忽略报错并继续执行**：跳过目标数据库中是否有同名表的检查项。 **警告** 选择为**忽略报错并继续执行**，可能导致数据不一致，给业务带来风险，例如： ■ 表结构一致的情况下，在目标库遇到与源库主键的值相同的记录，则会保留目标集群中的该条记录，即源库中的该条记录不会同步至目标数据库中。 ■ 表结构不一致的情况下，可能会导致无法初始化数据、只能同步部分列的数据或同步失败。
多表归并	○ 选择为**是**：通常在OLTP场景中，为提高业务表响应速度，通常会做分库分表处理。而在云原生数据仓库AnalyticDB PostgreSQL中单个数据表可存储海量数据，使用单表查询更加便捷。此类场景中，您可以借助DTS的多表归并功能将源库中多个表结构相同的表（即各分表）同步至云原生数据仓库AnalyticDB PostgreSQL中的同一个表中。 **说明** ■ 选择源库的多个表后，您需要通过对象名映射功能，将其改为云原生数据仓库AnalyticDB PostgreSQL中的同一个表名。关于对象名映射功能的介绍，请参见设置同步对象在目标实例中的名称。 ■ 您需要在云原生数据仓库AnalyticDB PostgreSQL的同步目标表中增加 `dts_data_source` 列（类型为text）来存储数据来源。DTS将以 `<dts数据同步实例ID>:<源数据库名>.<源Schema名>.<源表名>` 的格式写入列值用于区分表的来源，例如 `dts********:dtstestdata.testschema.customer1` 。 ■ 多表归并功能基于任务级别，即不支持基于表级别执行多表归并。如果需要让部分表执行多表归并，另一部分不执行多表归并，您需要创建两个数据同步作业。 ○ 选择为**否**：默认选项。
同步操作类型	根据业务选中需要同步的操作类型，默认情况下都处于选中状态。

配置	说明
选择同步对象	在**源库对象**框中单击待迁移的对象，然后单击 › 图标将其移动至**已选择对象**框。 同步对象的选择粒度为库、表。 **说明** • 默认情况下，同步对象的名称保持不变。如果您需要同步对象在目标实例上名称不同，请使用对象名映射功能，详情请参见设置同步对象在目标实例中的名称。 • 如果配置**多表归并**为**是**，在选择源库的多个表后，您需要通过对象名映射功能，将其改为云原生数据仓库AnalyticDB PostgreSQL中的同一个表名。
为目标对象添加引号	选择是否需要为目标对象名添加引号。如果选择为**是**，且存在下述情况，DTS在结构初始化阶段和增量数据迁移阶段会为目标对象添加单引号或双引号： • 源库所属的业务环境对大小写敏感且大小写混用。 • 源表名不是以字母开头，且包含字母、数字或特殊字符以外的字符。 **说明** 特殊字符仅支持下划线（_），井号（#）和美元符号（$）。 • 待迁移的Schema、表或列名称是目标库的关键字、保留字或非法字符。 **说明** 如果选择添加引号，在数据同步完成后，您需使用带引号的目标对象名进行查询。
映射名称更改	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。
源、目标库无法连接重试时间	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。 **说明** 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

9. 设置待同步的表在云原生数据仓库AnalyticDB PostgreSQL中表类型、主键列和分布键信息。

> **说明** 关于主键列和分布键的详细说明，请参见表的约束定义和表分布键定义。

10. 上述配置完成后，单击页面右下角的**预检查并启动**。

> **说明**
> - 在同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动同步作业。
> - 如果预检查失败，单击具体检查项后的，查看失败详情。
>   - 您可以根据提示修复后重新进行预检查。
>   - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项并重新进行预检查**，跳过告警检测项重新进行预检查。

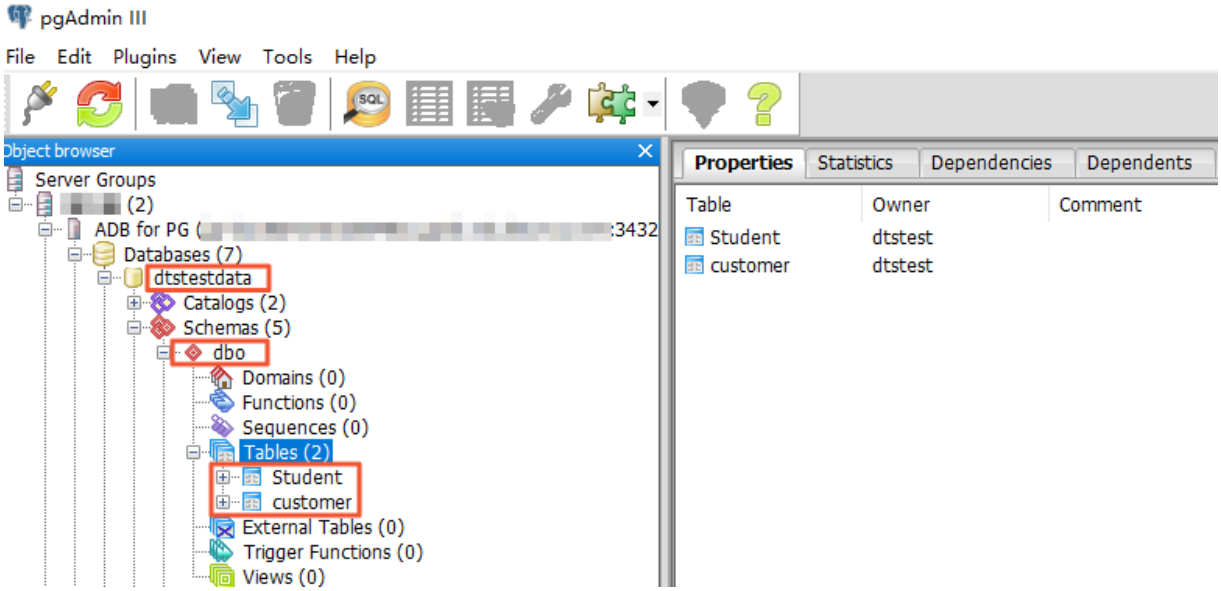
11. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。
12. 等待同步作业的链路初始化完成，直至处于**同步中**状态。

    您可以在**数据同步**页面，查看数据同步作业的状态。

常见问题

Q：如何在云原生数据仓库AnalyticDB PostgreSQL找到同步的目标表？

A：DTS的结构初始化会遵循源库的结构将其同步至目标库。本案例中，您可以在目标实例的 dtstestdata 数据库的 dbo Schema中，找到 customer 表和 Student 表，如下图所示。



10.2. RDS MySQL数据同步至AnalyticDB PostgreSQL版

数据传输服务DTS（Data Transmission Service）支持将RDS MySQL同步至AnalyticDB PostgreSQL版。通过DTS提供的数据同步功能，可以轻松实现数据的流转，将企业数据集中分析。

前提条件

- RDS MySQL中待同步的数据表必须具备主键。
- 已创建目标云原生数据仓库AnalyticDB PostgreSQL实例，如未创建请参见创建云原生数据仓库AnalyticDB PostgreSQL实例。

注意事项

DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。

同步限制

- 同步对象仅支持数据表。
- 不支持BIT、VARBIT、GEOMETRY、ARRAY、UUID、TSQUERY、TSVECTOR、TXID_SNAPSHOT类型的数据同步。
- 暂不支持同步前缀索引，如果源库存在前缀索引可能导致数据同步失败。
- 在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。

支持同步的SQL操作

- DML操作：INSERT、UPDATE、DELETE。

- DDL操作：ADD COLUMN。

> **说明** 不支持CREATE TABLE操作，如果您需要将新增的表作为同步对象，则需要执行新增同步对象操作。

支持的同步架构

- 1对1单向同步。
- 1对多单向同步。
- 多对1单向同步。

术语及概念对应关系

MySQL	云原生数据仓库AnalyticDB PostgreSQL
Database	Schema
Table	Table

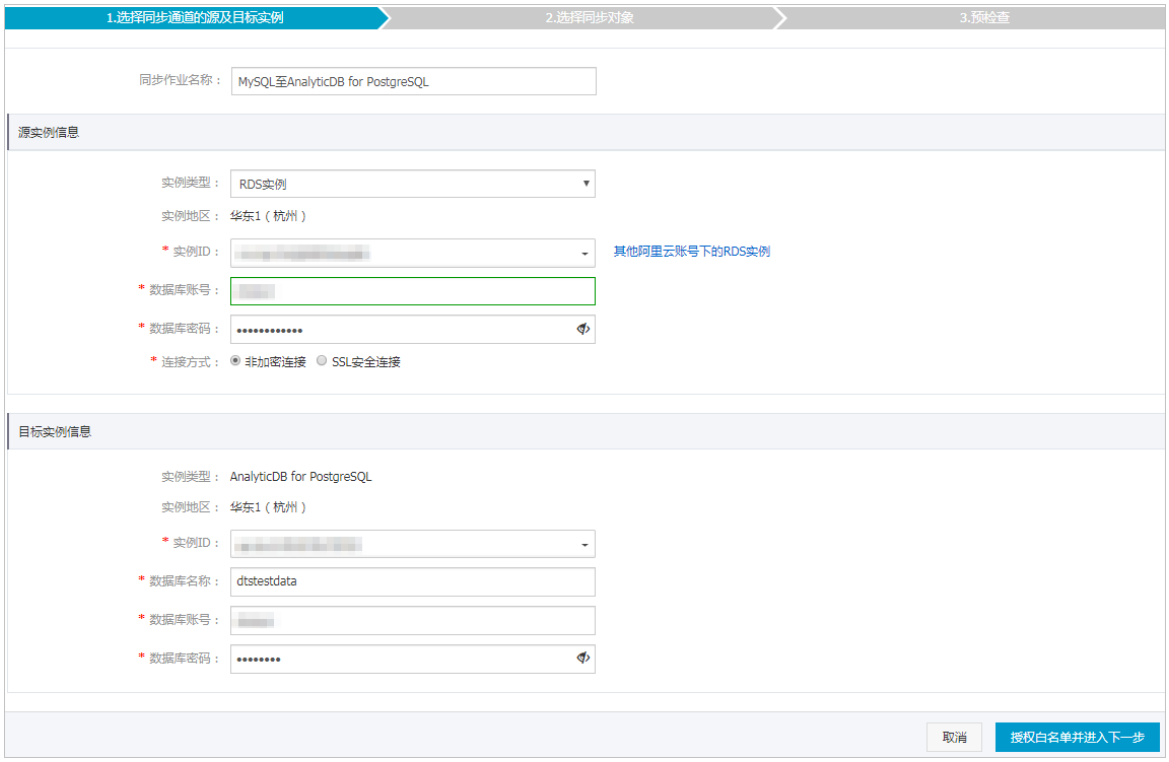
操作步骤

1. 购买数据同步作业，详情请参见购买流程。

   > **说明** 购买时，选择源实例为MySQL，目标实例为AnalyticDB for PostgreSQL，并选择同步拓扑为**单向同步**。

2. 登录数据传输控制台。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。

5. 定位至已购买的数据同步实例，单击**配置同步链路**。
6. 配置同步作业的源实例及目标实例信息。



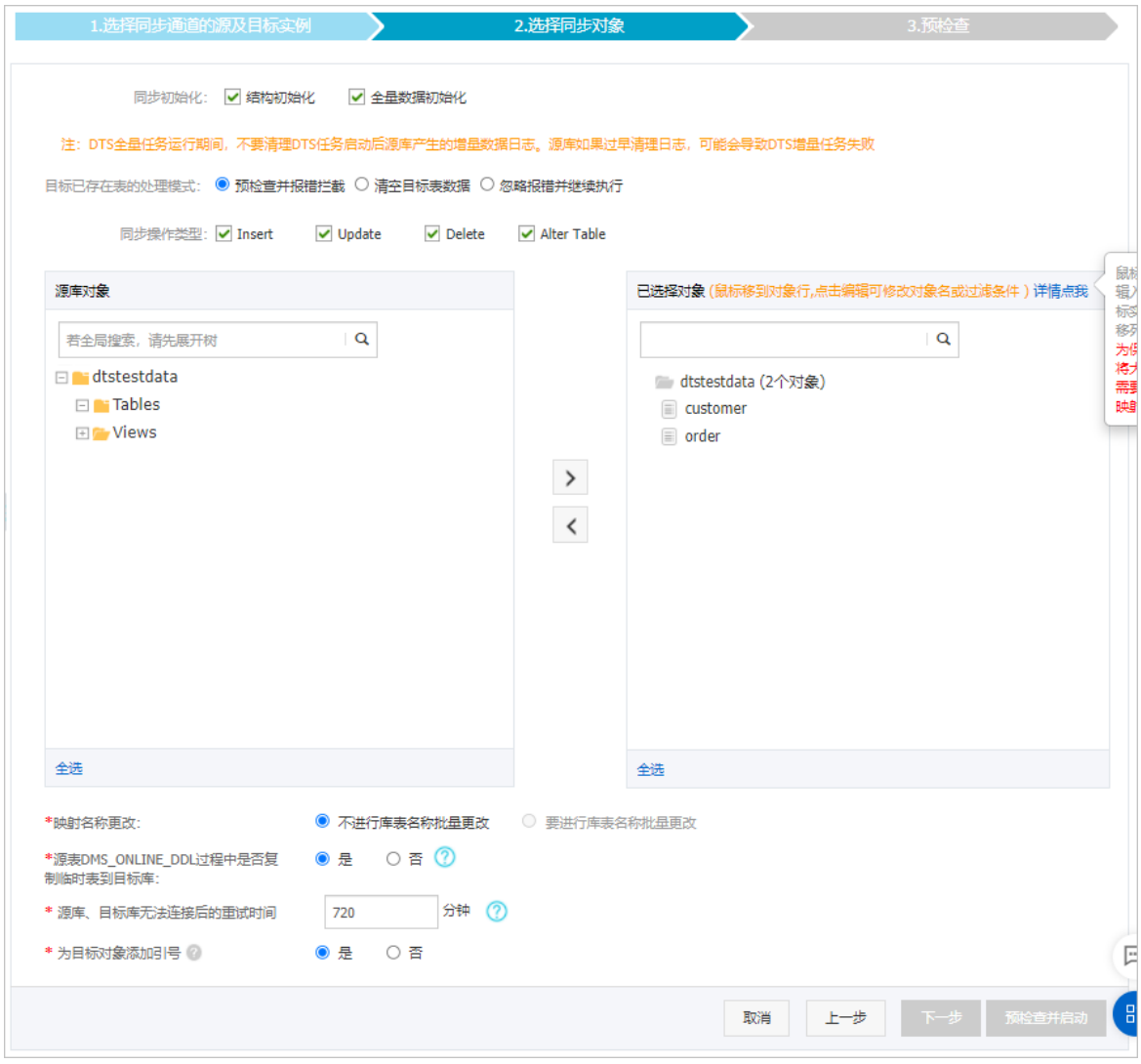
类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
源实例信息	实例类型	选择**RDS实例**。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	实例ID	选择源RDS实例ID。
	数据库账号	填入RDS MySQL的数据库账号。 **说明** 当源RDS实例的数据库类型为MySQL 5.5或MySQL 5.6时，没有**数据库账号**和**数据库密码**的配置选项。
	数据库密码	填入该数据库账号对应的密码。
	连接方式	根据需求选择**非加密连接**或**SSL安全连接**。如果设置为**SSL安全连接**，您需要提前开启RDS实例的SSL加密功能，详情请参见设置SSL加密。
	实例类型	固定为**AnalyticDB for PostgreSQL**，无需设置。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	实例ID	选择目标云原生数据仓库AnalyticDB PostgreSQL实例ID。
	数据库名称	填入同步目标表所属的数据库名称。

类别	配置	说明
目标实例信息	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL的**初始账号**，详情请参见创建数据库账号。 **说明** 您也可以填入具备RDS_SUPERUSER权限的账号，创建方法请参见用户权限管理。
	数据库密码	填入该数据库账号对应的密码。

7. 单击页面右下角的**授权白名单并进入下一步**。

   说明 此步骤会将DTS服务器的IP地址自动添加到RDS MySQL和云原生数据仓库AnalyticDB PostgreSQL的白名单中，用于保障DTS服务器能够正常连接源集群和目标实例。

8. 配置同步策略及同步对象。



类别	配置	说明
同步策略配置	同步初始化	默认情况下，您需要同时选中**结构初始化**和**全量数据初始化**。预检查完成后，DTS会将源实例中待同步对象的结构及数据在目标实例中初始化，作为后续增量同步数据的基线数据。
	目标已存在表的处理模式	○ **清空目标表的数据** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化之前将目标表的数据清空。适用于完成同步任务测试后的正式同步场景。 ○ **忽略报错并继续执行** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化时直接追加数据。适用于多张表同步到一张表的汇总同步场景。
	同步操作类型	根据业务需求选择需要同步的操作类型： ○ Insert ○ Update ○ Delete ○ AlterTable
选择同步对象	无	在**源库对象**框中单击待同步的表，然后单击 › 图标将其移动至**已选择对象**框。 **说明** ○ 同步对象的选择粒度为表。 ○ 如果需要目标表中的列名称与源表不同，需要使用DTS的字段映射功能，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	无	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。

类别	配置	说明
源表DMS_ONLINE_DDL过程中是否复制临时表到目标库	无	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 ○ **是**：同步Online DDL变更产生的临时表数据。 **说明** Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 ○ **否**：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。 **说明** 该方案会导致目标库锁表。
源、目标库无法连接重试时间	无	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。 **说明** 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

9. 设置待同步的表在云原生数据仓库AnalyticDB PostgreSQL中的主键列和分布列信息。

说明 当您在上一步中选择了**结构初始化**才会出现该页面。关于主键列和分布列的详细说明，请参见表的约束定义和表分布键定义。

10. 上述配置完成后，单击页面右下角的**预检查并启动**。

> **说明**
> - 在同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动同步作业。
> - 如果预检查失败，单击具体检查项后的，查看失败详情。
>   - 您可以根据提示修复后重新进行预检查。
>   - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项并重新进行预检查**，跳过告警检测项重新进行预检查。

11. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。
12. 等待同步作业的链路初始化完成，直至处于**同步中**状态。

    您可以在**数据同步**页面，查看数据同步作业的状态。

10.3. PolarDB MySQL数据同步至AnalyticDB PostgreSQL

数据传输服务DTS（Data Transmission Service）支持将PolarDB MySQL数据同步至AnalyticDB PostgreSQL，帮助您轻松实现数据的流转，将企业数据集中分析。

前提条件

- PolarDB MySQL集群已开启Binlog，详情请参见如何开启Binlog。
- PolarDB MySQL集群中待同步的数据表必须具备主键。
- 已创建目标云原生数据仓库AnalyticDB PostgreSQL实例，详情请参见创建云原生数据仓库AnalyticDB PostgreSQL实例。

注意事项

- DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。
- 全量初始化过程中，并发INSERT会导致目标实例的表碎片，全量初始化完成后，目标实例的表空间比源集群的表空间大。

同步限制

- 同步对象仅支持数据表。
- 不支持BIT、VARBIT、GEOMETRY、ARRAY、UUID、TSQUERY、TSVECTOR、TXID_SNAPSHOT类型的数据同步。
- 暂不支持同步前缀索引，如果源库存在前缀索引可能导致数据同步失败。

- 在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。

支持同步的SQL操作

- DML操作：INSERT、UPDATE、DELETE。
- DDL操作：ADD COLUMN。

> **说明** 不支持CREATE TABLE操作，如果您需要将新增的表作为同步对象，则需要执行新增同步对象操作。

支持的同步架构

- 1对1单向同步。
- 1对多单向同步。
- 多对1单向同步。

术语对应关系

PolarDB MySQL	云原生数据仓库AnalyticDB PostgreSQL
Database	Schema
Table	Table

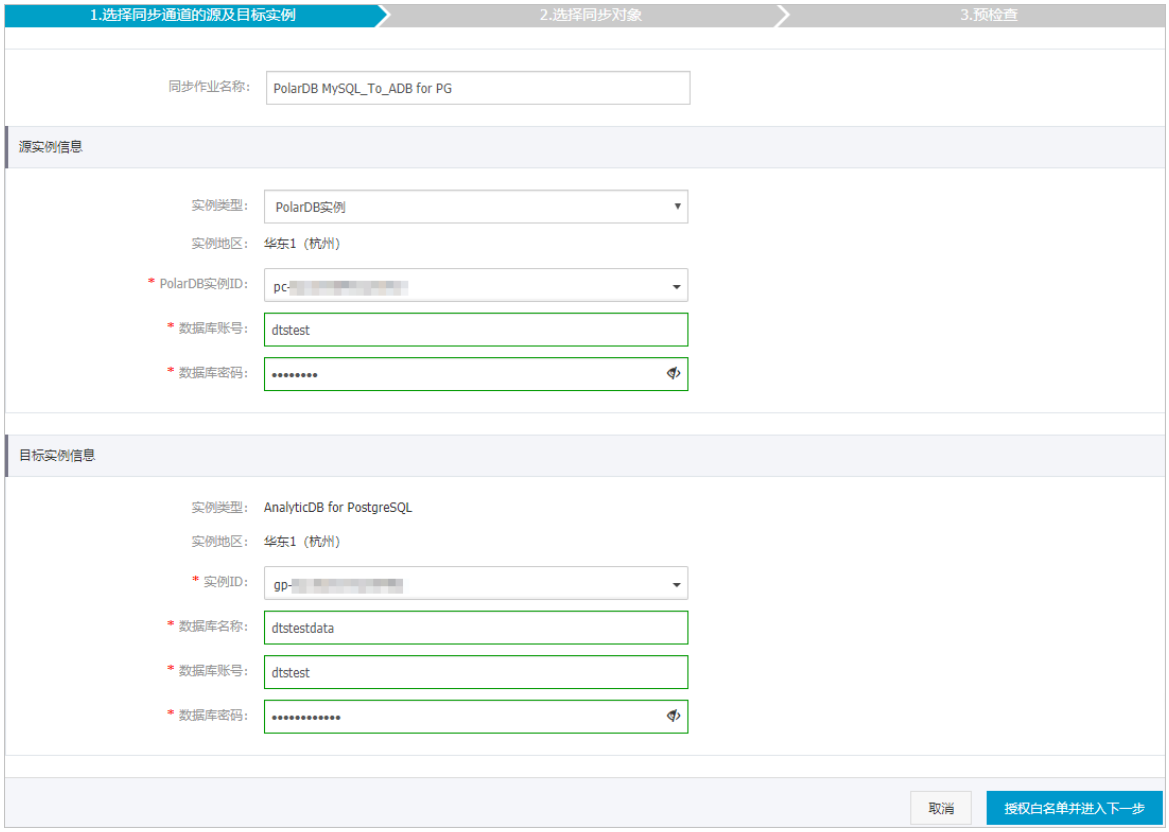
操作步骤

1. 购买数据同步作业，详情请参见购买流程。

   > **说明** 购买时，选择源实例为PolarDB、目标实例为AnalyticDB PostgreSQL，并选择同步拓扑为**单向同步**。

2. 登录数据传输控制台。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。

5. 定位至已购买的数据同步实例，单击**配置同步链路**。
6. 配置同步通道的源实例及目标实例信息。



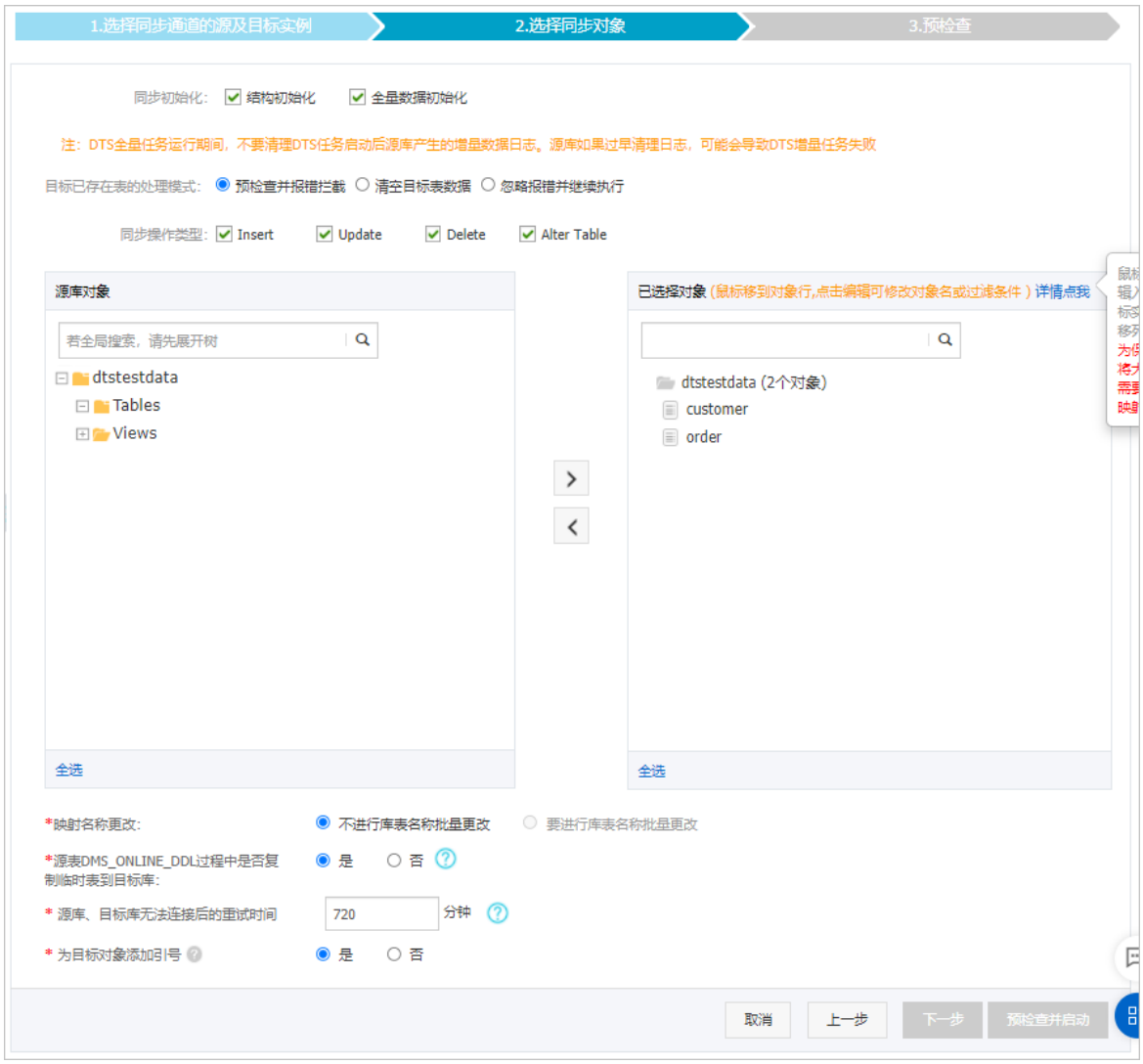
类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
源实例信息	实例类型	固定为**PolarDB实例**。
	实例地区	购买数据同步实例时选择的源PolarDB集群的地域信息，不可变更。
	PolarDB实例ID	选择PolarDB集群ID。
	数据库账号	填入PolarDB集群的数据库账号。 **说明** 该账号需具备待同步对象的读权限。
	数据库密码	填入该数据库账号的密码。
	实例类型	固定为**AnalyticDB for PostgreSQL**，无需设置。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	实例ID	选择云原生数据仓库AnalyticDB PostgreSQL实例ID。
	数据库名称	填入云原生数据仓库AnalyticDB PostgreSQL实例中，待同步的目标表所属的数据库名称。

目标实例信息类别	配置	说明
	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL的**初始账号**，详情请参见创建数据库账号。 **说明** 您也可以填入具备RDS_SUPERUSER权限的账号，创建方法请参见用户权限管理。
	数据库密码	填入数据库账号的密码。

7. 单击页面右下角的**授权白名单并进入下一步**。

   说明 此步骤会将DTS服务器的IP地址自动添加PolarDB MySQL和云原生数据仓库AnalyticDB PostgreSQL的白名单中，用于保障DTS服务器能够正常连接源集群和目标实例。

8. 配置同步策略及同步对象。



类别	配置	说明
同步策略配置	同步初始化	默认情况下，您需要同时选中**结构初始化**和**全量数据初始化**。预检查完成后，DTS会将源实例中待同步对象的结构及数据在目标实例中初始化，作为后续增量同步数据的基线数据。
	目标已存在表的处理模式	○ **清空目标表的数据** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化之前将目标表的数据清空。适用于完成同步任务测试后的正式同步场景。 ○ **忽略报错并继续执行** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化时直接追加数据。适用于多张表同步到一张表的汇总同步场景。
	同步操作类型	根据业务需求选择需要同步的操作类型： ○ Insert ○ Update ○ Delete ○ AlterTable
选择同步对象	无	在**源库对象**框中单击待同步的表，然后单击 › 图标将其移动至**已选择对象**框。 **说明** ○ 同步对象的选择粒度为表。 ○ 如果需要目标表中的列名称与源表不同，需要使用DTS的字段映射功能，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	无	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。

类别	配置	说明
源表DMS_ONLINE_DDL过程中是否复制临时表到目标库	无	如源库使用数据管理DMS（Data Management Service）执行Online DDL变更，您可以选择是否同步Online DDL变更产生的临时表数据。 ○ **是**：同步Online DDL变更产生的临时表数据。 **说明** Online DDL变更产生的临时表数据过大，可能会导致同步任务延迟。 ○ **否**：不同步Online DDL变更产生的临时表数据，只同步源库的原始DDL数据。 **说明** 该方案会导致目标库锁表。
源、目标库无法连接重试时间	无	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。 **说明** 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

9. 设置待同步的表在云原生数据仓库AnalyticDB PostgreSQL中的主键列和分布列信息。

说明 当您在上一步中选择了**结构初始化**才会出现该页面。关于主键列和分布列的详细说明，请参见表的约束定义和表分布键定义。

10. 上述配置完成后，单击页面右下角的**预检查并启动**。

> **说明**
> - 在同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动同步作业。
> - 如果预检查失败，单击具体检查项后的，查看失败详情。
>   - 您可以根据提示修复后重新进行预检查。
>   - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项并重新进行预检查**，跳过告警检测项重新进行预检查。

11. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。
12. 等待同步作业的链路初始化完成，直至处于**同步中**状态。

    您可以在**数据同步**页面，查看数据同步作业的状态。

10.4. 从RDS PostgreSQL同步至云原生数据仓库AnalyticDB PostgreSQL版

数据传输服务DTS（Data Transmission Service）支持将RDS PostgreSQL同步至AnalyticDB PostgreSQL版。通过DTS提供的数据同步功能，可以轻松实现数据的流转，将企业数据集中分析。

前提条件

- RDS PostgreSQL中待同步的数据表必须具备主键。
- 已创建目标云原生数据仓库AnalyticDB PostgreSQL实例，如未创建请参见创建云原生数据仓库AnalyticDB PostgreSQL实例。

注意事项

- 一个数据同步作业只能同步一个数据库，如果有多个数据库需要同步，则需要为每个数据库创建数据同步作业。
- 在数据同步的过程中，如果要将源库中创建的新表作为同步对象，您需要对该表执行如下操作以保障该表数据同步的一致性。

  ```
  ALTER TABLE schema.table REPLICA IDENTITY FULL;
  ```

- 为保障同步任务的正常进行，目前仅支持RDS PostgreSQL 11进行主备切换，且需设置参数 `rds_failover_slot_mode` 为 `sync` ，设置方式，请参见逻辑复制槽故障转移（Logical Replication Slot Failover）。

  > **警告** 自建PostgreSQL和其他版本的RDS PostgreSQL进行主备切换，会导致同步中断。

同步限制

- 不支持结构初始化，即不支持将源库中待同步对象的结构定义（例如表结构）同步至目标库中。

- 同步对象仅支持数据表。
- 不支持BIT、VARBIT、GEOMETRY、ARRAY、UUID、TSQUERY、TSVECTOR、TXID_SNAPSHOT类型的数据同步。
- 同步过程中，如果对源库中的同步对象执行了DDL操作，需要手动在目标库中执行对应的DDL操作，然后重启数据同步作业。

支持的同步语法

仅支持INSERT、UPDATE、DELETE。

准备工作

1. 调整源RDS实例的 `wal_level` 参数设置。

   > **警告** 修改 `wal_level` 参数后需要重启实例才能生效，请评估对业务的影响，在业务低峰期进行修改。

   i. 登录RDS管理控制台。
   ii. 在页面左上角，选择实例所在地域。
   iii. 找到目标实例，单击实例ID。
   iv. 在左侧导航栏，单击**参数设置**。
   v. 在**参数设置**页面找到 `wal_level` 参数，将参数值改为 `logical` 。
2. 根据源RDS实例中待同步对象的结构，在目标云原生数据仓库AnalyticDB PostgreSQL中创建相应的数据库、Schema、表等结构信息，详情请参见SQL语法。

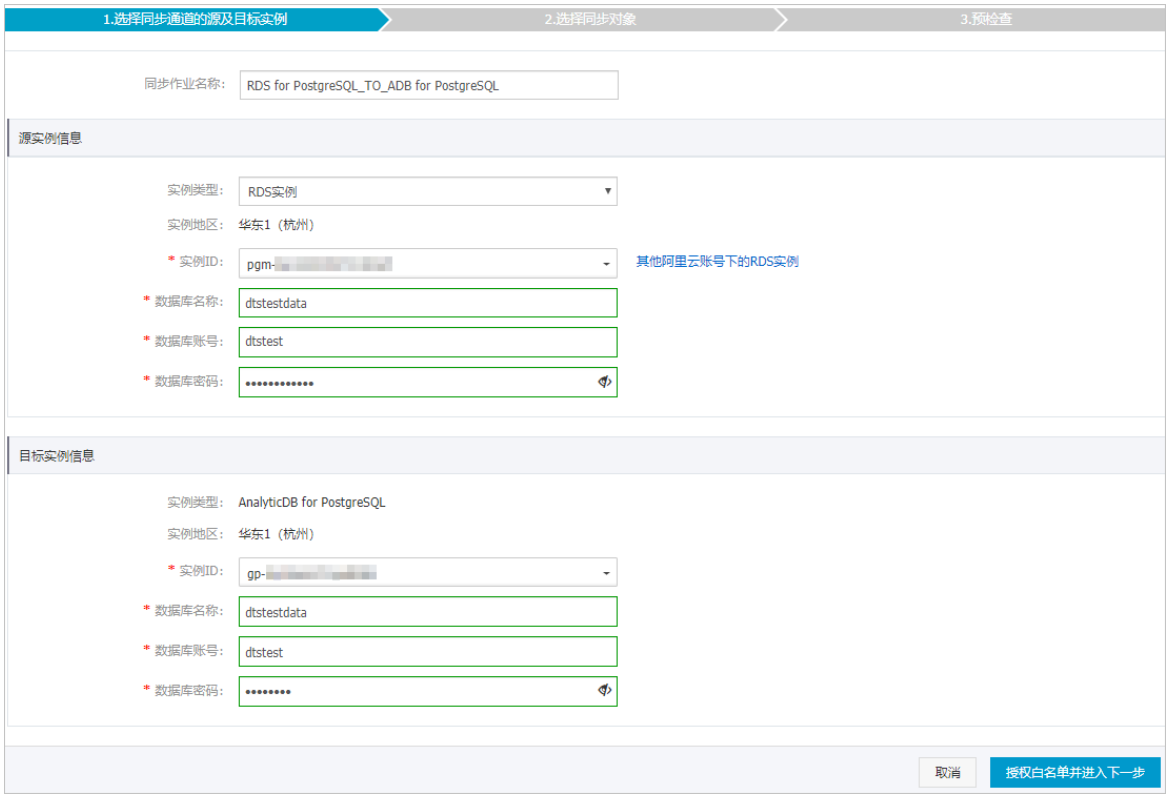
操作步骤

1. 购买数据同步作业，详情请参见购买流程。

   > **说明** 购买时，选择源实例为PostgreSQL、目标实例为AnalyticDB for PostgreSQL，并选择同步拓扑为**单向同步**。

2. 登录数据传输控制台。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。

5. 定位至已购买的数据同步实例，单击**配置同步链路**。
6. 配置同步作业的源实例及目标实例信息。



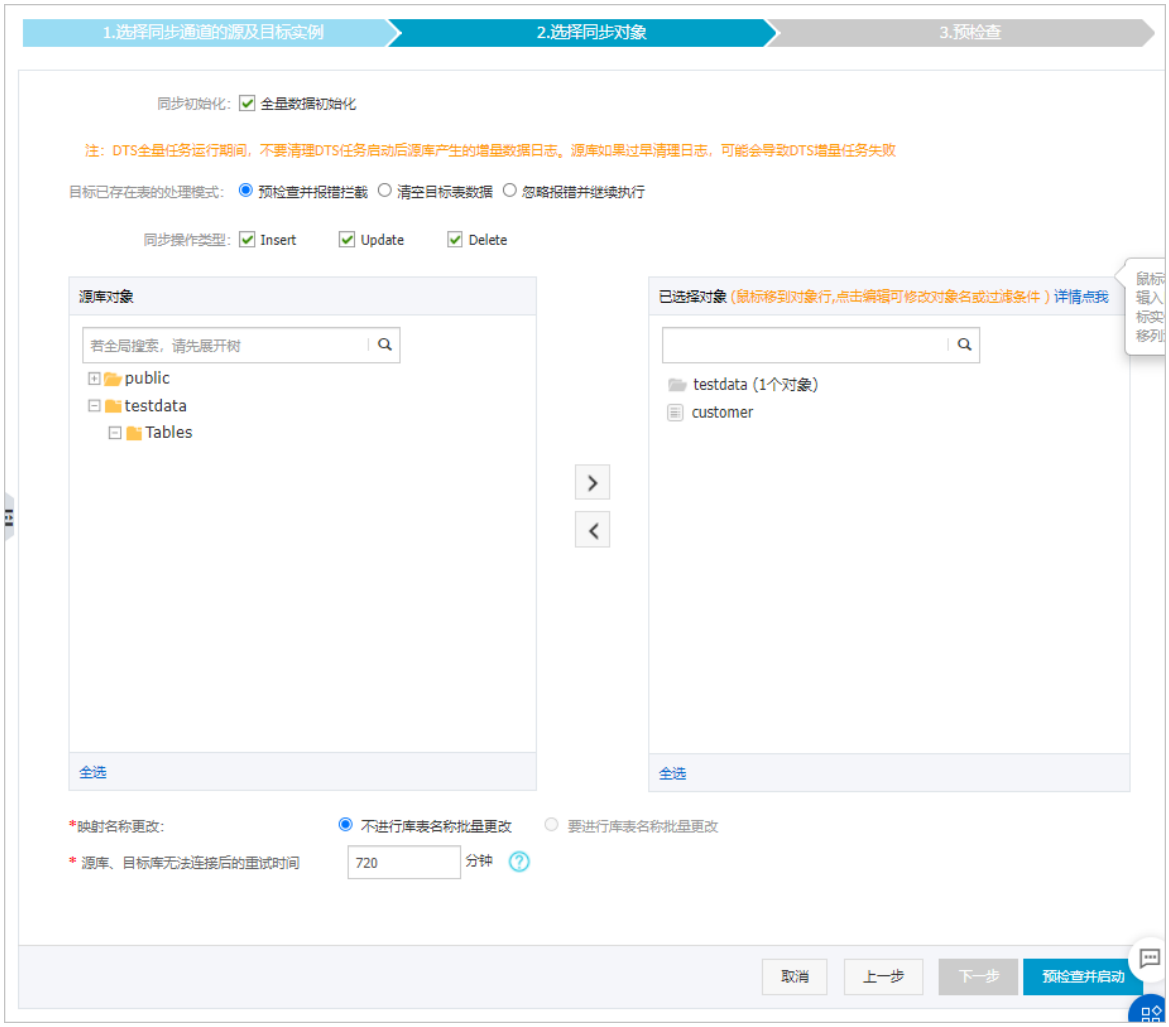
类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
源实例信息	实例类型	选择**RDS实例**。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	实例ID	选择RDS PostgreSQL实例ID。
	数据库名称	填入待同步的表所属的数据库名称。
	数据库账号	填入RDS PostgreSQL的数据库账号，需具备superuser权限。 **说明** 当源库为RDS PostgreSQL 9.4，且仅需同步DML操作时，账号具备replication权限即可。
	数据库密码	填入该数据库账号对应的密码。
	实例类型	固定为**AnalyticDB for PostgreSQL**，无需设置。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	实例ID	选择云原生数据仓库AnalyticDB PostgreSQL实例ID。

类别	配置	说明
目标实例信息	数据库名称	填入同步目标表所属的数据库名称。 **说明** 该库须在云原生数据仓库AnalyticDB PostgreSQL中存在，如不存在请创建数据库。
	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL的**初始账号**，详情请参见创建数据库账号。 **说明** 您也可以填入具备RDS_SUPERUSER权限的账号，创建方法请参见用户权限管理。
	数据库密码	填入该数据库账号对应的密码。

7. 单击页面右下角的**授权白名单并进入下一步**。

   说明 此步骤会将DTS服务器的IP地址自动添加到RDS PostgreSQL和云原生数据仓库AnalyticDB PostgreSQL的白名单中，用于保障DTS服务器能够正常连接源和目标实例。

8. 配置同步策略及对象信息。



类别	配置	说明
同步策略配置	同步初始化	默认情况下，您需要勾选**全量数据初始化**。预检查完成后，DTS会将源实例中待同步对象的存量数据同步至目标实例，作为后续增量同步数据的基线数据。
	目标已存在表的处理模式	○ **清空目标表的数据** 在预检查阶段跳过**目标表是否为空**的检查项目。全量初始化之前将目标表的数据清空。适用于完成同步任务测试后的正式同步场景。 ○ **忽略报错并继续执行** 在预检查阶段跳过**目标表是否为空**的检查项目。全量初始化时直接追加迁移数据。适用于多张表同步到一张表的汇总同步场景。

类别	配置	说明
	同步操作类型	根据业务需求选择需要同步的操作类型： **说明** 不支持AlterTable。 ○ Insert ○ Update ○ Delete ○ AlterTable
选择同步对象	无	在**源库对象**框中单击待同步的表，然后单击 › 将其移动至**已选择对象**框。 **说明** ○ 同步对象的选择粒度为表。 ○ 如果需要目标表中的列名称与源表不同，则需要使用DTS的字段映射功能，详情请参见设置同步对象在目标实例中的名称。
映射名称更改	无	如需更改同步对象在目标实例中的名称，请使用对象名映射功能，详情请参见库表列映射。
源、目标库无法连接重试时间	无	当源、目标库无法连接时，DTS默认重试720分钟（即12小时），您也可以自定义重试时间。如果DTS在设置的时间内重新连接上源、目标库，同步任务将自动恢复。否则，同步任务将失败。 **说明** 由于连接重试期间，DTS将收取任务运行费用，建议您根据业务需要自定义重试时间，或者在源和目标库实例释放后尽快释放DTS实例。

9. 上述配置完成后，单击页面右下角的**预检查并启动**。

说明

- 在同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动同步作业。
- 如果预检查失败，单击具体检查项后的 ⓘ，查看失败详情。
  - 您可以根据提示修复后重新进行预检查。
  - 如无需修复告警检测项，您也可以选择**确认屏蔽**、**忽略告警项并重新进行预检查**，跳过告警检测项重新进行预检查。

10. 在**预检查**对话框中显示**预检查通过**后，关闭**预检查**对话框，同步作业将正式开始。
11. 等待同步作业的链路初始化完成，直至处于**同步中**状态。

    您可以在**数据同步**页面，查看数据同步作业的状态。

10.5. rds_dbsync迁移/同步MySQL数据到AnalyticDB for PostgreSQL

rds_dbsync为开源的数据同步迁移工具，其mysql2pgsql功能支持不落地的把MySQL中的表迁移到AnalyticDB PostgreSQL/Greenplum Database/PostgreSQL/PPAS。此工具的原理是，同时连接源端MySQL数据库和目的端数据库，从MySQL库中通过查询得到要导出的数据，然后通过 COPY命令导入到目的端。此工具支持多线程导入（每个工作线程负责导入一部分数据库表）。

参数配置

修改配置文件my.cfg、配置源和目的库连接信息。

- 源库MySQL的连接信息如下：

  > 注意　源库MySQL的连接信息中，用户需要有对所有用户表的读权限。

  ```
  [src.mysql]
  host = "192.168.1.1"
  port = "3306"
  user = "test"
  password = "test"
  db = "test"
  encodingdir = "share"
  encoding = "utf8"
  ```

- 目的库pgsql（包括Postgresql、PPAS和AnalyticDB PostgreSQL）的连接信息如下：

  > 注意　目的库pgsql的连接信息，用户需要对目标表有写的权限。

  ```
  [desc.pgsql]
  connect_string = "host=192.168.1.2 dbname=test port=3432  user=test password=pgsql"
  ```

mysql2pgsql用法

mysql2pgsql的用法如下所示：

```
./mysql2pgsql -l <tables_list_file> -d -n -j <number of threads> -s <schema of target table>
```

参数说明：

- -l：可选参数，指定一个文本文件，文件中含有需要同步的表；如果不指定此参数，则同步配置文件中指定数据库下的所有表。 <tables_list_file> 为一个文件名，里面含有需要同步的表集合以及表上查询的条件，其内容格式示例如下：

```
table1 : select * from table_big where column1 < '2016-08-05'
table2 :
table3
table4: select column1, column2 from tableX where column1 != 10
table5: select * from table_big where column1 >= '2016-08-05'
```

- -d：可选参数，表示只生成目的表的建表DDL语句，不实际进行数据同步。
- -n：可选参数，需要与-d一起使用，指定在DDL语句中不包含表分区定义。
- -j：可选参数，指定使用多少线程进行数据同步；如果不指定此参数，会使用5个线程并发。
- -s：可选参数，指定目标表的schema，目前仅支持设定为public。

典型用法

全库迁移

全库迁移的操作步骤如下所示：

1. 通过如下命令，获取目的端对应表的DDL。

```
./mysql2pgsql -d
```

2. 根据这些DDL，再加入Distribution Key等信息，在目的端创建表。
3. 执行如下命令，同步所有表：

```
./mysql2pgsql
```

此命令会把配置文件中所指定数据库中的所有MySQL 表数据迁移到目的端。过程中使用5个线程（即缺省线程数为5），读取和导入所有涉及的表数据。

部分表迁移

1. 编辑一个新文件tab_list.txt，放入如下内容：

```
t1
t2 : select * from t2 where c1 > 138888
```

2. 执行如下命令，同步指定的t1和t2表（注意t2表只迁移符合c1 > 138888条件的数据）：

```
./mysql2pgsql -l tab_list.txt
```

下载与说明

- 下载mysql2pgsql二进制安装包。
- 查看mysql2pgsql源码编译说明。

10.6. rds_dbsync迁移/同步PostgreSQL数据到AnalyticDB PostgreSQL

开源工具 rds_dbsync的pgsql2pgsql功能，支持把AnalyticDB PostgreSQL、Greenplum Database、PostgreSQL、PPAS中的表迁AnalyticDB PostgreSQL、Greenplum Database、PostgreSQL、PPAS。

pgsql2pgsql支持的功能

pgsql2pgsql支持如下功能：

- PostgreSQL、PPAS、Greenplum Database、AnalyticDB PostgreSQL全量数据迁移到PostgreSQL、PPAS、Greenplum Database、AnalyticDB PostgreSQL。
- PostgreSQL或PPAS（版本大于9.4）全量+增量迁移到PostgreSQL或PPAS。

参数配置

修改配置文件postgresql.conf、配置源和目的库连接信息。

- 源库pgsql连接信息如下所示：

  > **注意** 源库pgsql的连接信息中，用户最好是对应DB的owner。

  ```
  [src.pgsql]
  connect_string = "host=192.168.1.1 dbname=test port=3432 user=test password=pgsql"
  ```

- 本地临时Database pgsql连接信息如下所示：

  ```
  [local.pgsql]
  connect_string = "host=192.168.1.2 dbname=test port=3432 user=test2 password=pgsql"
  ```

- 目的库pgsql连接信息如下所示：

  > **注意** 目的库pgsql的连接信息，用户需要对目标表有写权限。

  ```
  [desc.pgsql]
  connect_string = "host=192.168.1.3 dbname=test port=3432 user=test3 password=pgsql"
  ```

> **注意**
>
> - 如果要做增量数据同步，连接源库需要有创建replication slot的权限。
> - 由于PostgreSQL 9.4及以上版本支持逻辑流复制，所以支持作为数据源的增量迁移。打开下列内核参数才能让内核支持逻辑流复制功能。
>
>   ```
>   wal_level = logical
>   max_wal_senders = 6
>   max_replication_slots = 6
>   ```

pgsql2pgsql用法

全库迁移

进行全库迁移，请执行如下命令：

```
./pgsql2pgsql
```

迁移程序会默认把对应pgsql库中所有用户的表数据将迁移到pgsql。

状态信息查询

连接本地临时Database，可以查看到单次迁移过程中的状态信息。这些信息被放在表db_sync_status中，包括全量迁移的开始和结束时间、增量迁移的开始时间和增量同步的数据情况。

下载与说明

- 下载rds_dbsync二进制安装包。
- 查看rds_dbsync源码编译说明。